AFRL-HE-WP-TR-2004-0119

United States Air Force Research Laboratory

Aircraft Battle Damage Assessment and Repair (ABDAR) Evaluation

James Leftwich

GRACAR Corporation
714 Monument Avenue
Suite 201
Dayton, OH 45402

Arthur P. Grafton
Air Force Research Laboratory

September 2003

Final Report for the Period March 2003 to September 2003

Human Effectiveness Directorate
Deployment and Sustainment Division
Logistics Readiness Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

Approved for public release; distribution is unlimited.

NOTICES

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Federal Government agencies registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Rd., Ste 0944
Ft. Belvoir, VA 22060-6218

DISCLAIMER

This Technical Report is published as received and has not been edited by the Air Force Research Laboratory, Human Effectiveness Directorate.

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2004-0119

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

// Signed //

MARK M. HOFFMAN
Deputy Chief
Deployment and Sustainment Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2003	Final - March 2003 - September 2003

4. TITLE AND SUBTITLE
Aircraft Battle Damage Assessment and Repair (ABDAR) Evaluation

5. FUNDING NUMBERS
C: F33600-03-F-6065
PE: 62202F
PR: 1710
TA: D2
WU: 16

6. AUTHOR(S)
James Leftwich, Arthur P. Grafton

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
GRACAR Corporation
714 Monument Avenue
Suite 201
Dayton, OH 45402

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory, Human Effectiveness Directorate
Deployment and Sustainment Division
Air Force Materiel Command
Logistics Readiness Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-HE-WP-TR-2004-0119

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Aircraft Battle Damage Assessment and Repair (ABDAR) program is an advanced research and development (R&D) project under the sponsorship of Air Force Research Laboratory/Deployment and Sustainment Division Logistics Readiness Branch (AFRL/HESR). The intent of this system is to provide a significant enhancement in the capability of USAF Aircraft Battle Damage Repair (ABDR) assessors and technicians to rapidly assess battle damaged aircraft. These individuals face the critical task of assessing, repairing, and returning battle-damaged aircraft to mission readiness as rapidly as possible while maintaining a high level of precision. AFRL completed an ABDAR project in 2000 that featured prototype software that demonstrated how software and digitized technical data could increase speed and accuracy of the ABDR process. ABDR experts from AFMC/MSG also completed an ABDAR software tool in 2002, referred to in this document as the MSG mockup. AFMC/MSG and the Combat Logistics Support Squadrons (CLSS) have shown continued interest in this capability, but it is unknown what portions of the mockup software packages, if any can be used to produce an ABDAR system that meets minimum CLSS requirements and can be fielded quickly. The objective of the ABDAR evaluation effort was to assess the AFRL and MSG software currently demonstrating the concept of automating the documentation and repair of aircraft battle damage. The evaluation will determine what existing software code and processes, if any, are economically and technically valuable in developing a new product that meets CLSS requirements. In addition, this software will be evaluated to determine the level of effort required to modify them in order to meet the minimal functional requirements.

14. SUBJECT TERMS
Aircraft Battle Damage Assessment and Repair (ABDAR) Battle Damage Assessment
Aircraft Battle Damage Repair (ABDR) Battle Damage Repair Battle Repair

15. NUMBER OF PAGES
104

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

THIS PAGE INTENTIONALLY LEFT BLANK

PREFACE

The research documented in this technical report for the Aircraft Battle Damage Assessment and Repair (ABDAR) program was sponsored by the Air Force Research Laboratory, Human Effectiveness Directorate, Logistics Readiness Branch (AFRL/HESR), Wright-Patterson Air Force Base, OH. GRACAR Corporation performed the work under contract F33600-03-F-6065. Captain Arthur P. Grafton (AFRL/HESR) was the program manager for the effort.

CONTENTS

1. Introduction 1
   1.1 Background 1
   1.2 Scope 1
2. References 3
3. Requirements Traceability Matrix 4
4. Software Estimation Methodology 5
   4.1 Background 5
   4.2 Calibration 5
      4.2.1 Calibration by Project Type 5
      4.2.2 Calibration by Productivity Drivers 6
      4.2.3 Historical Data 6
   4.3 Goal Seeking 7
   4.4 Probability of Success 7
   4.5 Estimation Algorithms 7
      4.5.1 SLIM 8
      4.5.2 Cocomo 2.0 8
      4.5.3 Monte Carlo Simulation 8
   4.6 Construx Estimate Refinement 8
   4.7 Cost Estimation 8
   4.8 Basis of Estimation 9
5. System Evaluation 10
   5.1 Background 10
   5.2 Approach 10
      5.2.1 Software Quality Criteria 10
      5.2.2 ABDAR Implementation Criteria matrix 11
      5.2.3 Calculation of Final Grades 12
   5.3 Software Evaluation 12
      5.3.1 MSG 12
      5.3.2 AFRL 13
   5.4 Evaluation Conclusion 15
6. Proposed Solution 16
   6.1 Discussion 16
   6.2 Solution 16
      6.2.1 Requirements 16
      6.2.2 ABDAR Application Packages 16
   6.3 Size/Cost Estimations 17
      6.3.1 Package 1 – 33 Classes 18
      6.3.2 Package 2 – 45 Classes 18
      6.3.3 Package 3 – 391 Classes 19
      6.3.4 Package 4 – 213 Classes 20
      6.3.5 Package 5 - 83 Classes 20
   6.4 Estimation Quality 21
      6.4.1 Calibration Evaluation 21
      6.4.2 Scope Evaluation 21
      6.4.3 Phase Evaluation 21
      6.4.4 Suitability Evaluation 22
   6.5 Reusability 22
   6.6 Risks 22
7. Certification and Accreditation (C&A) 24

7.1 Background 24
7.2 Command, Control, Communications, Computer and Intelligence Support Plan 24
7.3 System Security Authorization Agreement (SSAA) 24
7.4 Certificate of Networthiness (CON) 24
8. Conclusion 25
8.1 Analysis of Existing AFRL/MSG Software 25
8.2 Estimation of Effort to Develop a New System 25
8.3 Recommendation 26
9. Notes 27
9.1 Acronyms 27
10. Appendix A: ABDAR Software Estimate – Package 1 30
10.1 Estimate Summary 30
10.1.1 Nominal Plan 30
10.1.2 Optimum Plan 30
10.2 Estimate Quality 31
10.2.1 Summary of Estimate Quality 31
10.2.2 Calibration Evaluation 31
10.2.3 Scope Evaluation 31
10.2.4 Phase Evaluation 31
10.2.5 Consistency Check 32
10.2.6 Suitability Evaluation 32
10.3 Planning Options Overview 33
10.3.1 Nominal Plan 33
10.3.2 Optimum Plan 33
10.3.3 Shortest-Schedule Plan 33
10.3.4 Least-Cost Plan 34
10.4 Priorities 34
10.5 Scope Probabilities 34
10.6 Effort Probabilities 35
11. Appendix B: ABDAR Software Estimate – Package 2 37
11.1 Estimate Summary 37
11.1.1 Nominal Plan 37
11.1.2 Optimum Plan 37
11.2 Estimate Quality 38
11.2.1 Summary of Estimate Quality 38
11.2.2 Calibration Evaluation 38
11.2.3 Scope Evaluation 38
11.2.4 Phase Evaluation 38
11.2.5 Consistency Check 38
11.2.6 Suitability Evaluation 39
11.3 Planning Options Overview 39
11.3.1 Nominal Plan 39
11.3.2 Optimum Plan 40
11.3.3 Shortest Schedule Plan 40
11.3.4 Least Cost Plan 40
11.4 Priorities 41
11.5 Scope Probabilities 41
11.6 Effort Probabilities 42
12. Appendix C: ABDAR Software Estimate – Package 3 43
12.1 Estimate Summary 43

12.1.1 Nominal Plan ... 43
12.1.2 Optimum Plan ... 43
12.2 Estimate Quality ... 44
12.2.1 Summary of Estimate Quality ... 44
12.2.2 Calibration Evaluation ... 44
12.2.3 Scope Evaluation ... 44
12.2.4 Phase Evaluation ... 44
12.2.5 Consistency Check ... 44
12.2.6 Suitability Evaluation ... 45
12.3 Planning Options Overview ... 45
12.3.1 Nominal Plan ... 45
12.3.2 Optimum Plan ... 46
12.3.3 Shortest Schedule Plan ... 46
12.3.4 Least Cost Plan ... 46
12.4 Priorities ... 47
12.5 Scope Probabilities ... 47
12.6 Effort Probabilities ... 48
13. Appendix D: ABDAR Software Estimate – Package 4 ... 50
13.1 Estimate Summary ... 50
13.1.1 Nominal Plan ... 50
13.1.2 Optimum Plan ... 50
13.2 Estimate Quality ... 51
13.2.1 Summary of Estimate Quality ... 51
13.2.2 Calibration Evaluation ... 51
13.2.3 Scope Evaluation ... 51
13.2.4 Phase Evaluation ... 51
13.2.5 Consistency Check ... 51
13.2.6 Suitability Evaluation ... 52
13.3 Planning Options Overview ... 52
13.3.1 Nominal Plan ... 52
13.3.2 Optimum Plan ... 53
13.3.3 Shortest Schedule Plan ... 53
13.3.4 Least Cost Plan ... 53
13.4 Priorities ... 54
13.5 Scope Probabilities ... 54
13.6 Effort Probabilities ... 55
14. Appendix E: ABDAR Software Estimate – Package 5 ... 57
14.1 Estimate Summary ... 57
14.1.1 Nominal Plan ... 57
14.1.2 Optimum Plan ... 57
14.2 Estimate Quality ... 58
14.2.1 Summary of Estimate Quality ... 58
14.2.2 Calibration Evaluation ... 58
14.2.3 Scope Evaluation ... 58
14.2.4 Phase Evaluation ... 58
14.2.5 Consistency Check ... 58
14.2.6 Suitability Evaluation ... 59
14.3 Planning Options Overview ... 60
14.3.1 Nominal Plan ... 60
14.3.2 Optimum Plan ... 60

14.3.3 Shortest-Schedule Plan....60
14.3.4 Least-Cost Plan....61
14.4 Priorities....61
14.5 Scope Probabilities....61
14.6 Effort Probabilities....62
15. Appendix F: Software Evaluation Criteria....64
15.1 Software Quality Criteria....64
15.2 Software Quality Method....70
16. Appendix G: ABDAR Software Implementation Criteria....71
17. Appendix H: Requirements Traceability Matrix....80
17.1 Package 1....80
17.2 Package 2....83
17.3 Package 3....85
17.4 Package 4....90
17.5 Package 5....95

TABLES

Table 4-1: Estimated Labor Rates....9
Table 4-2: Class Estimate....9
Table 5-1: Evaluation Grade Values....11
Table 5-2: Software Quality Criteria Weighting....12
Table 5-3: MSG Mockup Report Card....13
Table 5-4: AFRL Prototype Report Card....14
Table 6-1: Package 1 Nominal Plan....18
Table 6-2: Package 1 Optimum Plan....18
Table 6-3: Package 2 Nominal Plan....18
Table 6-4: Package 2 Optimum Plan....19
Table 6-5: Package 3 Nominal Plan....19
Table 6-6: Package 3 Optimum Plan....19
Table 6-7: Package 4 Nominal Plan....20
Table 6-8: Package 4 Optimum Plan....20
Table 6-9: Package 5 Nominal Plan....20
Table 6-10: Package 5 Optimum Plan....21
Table 6-11: Estimation Quality....22
Table 8-1: Software Package Estimates....26
Table 10-1: Package 1 Nominal Estimate....30
Table 10-2: Package 1 Optimum Estimate....30
Table 10-3: Package 1 Consistency Check....32
Table 10-4: Package 1 Planning - Nominal....33
Table 10-5: Package 1 Planning – Optimum....33
Table 10-6: Package 1 Planning – Shortest Schedule....33
Table 10-7: Package 1 Planning-Least Cost....34
Table 10-8: Package 1 Priorities....34
Table 10-9: Package 1 Scope Probabilities....35
Table 10-10: Package 1 Effort Probabilities....36
Table 11-1: Package 2 Nominal Estimate....37
Table 11-2: Package 2 Optimum Estimate....37
Table 11-3: Package 2 Consistency Check....39
Table 11-4: Package 2 Planning - Nominal....39

Table 11-5: Package 2 Planning - Optimum 40
Table 11-6: Package 2 Planning – Shortest Schedule 40
Table 11-7: Package 2 Planning – Least Cost 40
Table 11-8: Package 2 Priorities 41
Table 11-9: Package 2 Scope Probabilities 41
Table 11-10: Package 2 Effort Probabilities 42
Table 12-1: Package 3 Nominal Estimate 43
Table 12-2: Package 3 Optimum Estimate 43
Table 12-3: Package 3 Consistency Check 45
Table 12-4: Package 3 Planning - Nominal 45
Table 12-5: Package 3 Planning - Optimum 46
Table 12-6: Package 3 Planning – Shortest Schedule 46
Table 12-7: Package 3 Planning – Least Cost 46
Table 12-8: Package 3 Priorities 47
Table 12-9: Package 3 Scope Probabilities 48
Table 12-10: Package 3 Effort Probabilities 49
Table 13-1: Package 4 Nominal Estimate 50
Table 13-2: Package 4 Optimum Estimate 50
Table 13-3: Package 4 Consistency Check 52
Table 13-4: Package 4 Planning - Nominal 52
Table 13-5: Package 4 Planning - Optimum 53
Table 13-6: Package 4 Planning – Shortest Schedule 53
Table 13-7: Package 4 Planning – Least Cost 53
Table 13-8: Package 4 Priorities 54
Table 13-9: Package 4 Scope Probabilities 55
Table 13-10: Package 4 Effort Probabilities 56
Table 14-1: Package 5 Nominal Estimate 57
Table 14-2: Package 5 Optimum Estimate 57
Table 14-3: Package 5 Consistency Check 59
Table 14-4: Package 5 Planning - Nominal 60
Table 14-5: Package 5 Planning - Optimum 60
Table 14-6: Package 5 Planning – Shortest Schedule 60
Table 14-7: Package 5 Planning – Least Cost 61
Table 14-8: Package 5 Priorities 61
Table 14-9: Package 5 Scope Probabilities 62
Table 14-10: Package 5 Effort Probabilities 63

1. Introduction

The Aircraft Battle Damage Assessment and Repair (ABDAR) program is an advanced research and development (R&D) project under the sponsorship of Air Force Research Laboratory/Deployment and Sustainment Division Logistics Readiness Branch (AFRL/HESR). The intent of this system is to provide a significant enhancement in the capability of USAF Aircraft Battle Damage Repair (ABDR) assessors and technicians to rapidly assess battle damaged aircraft. These individuals face the critical task of assessing, repairing, and returning battle-damaged aircraft to mission readiness as rapidly as possible while maintaining a high level of precision. The specific objective of the ABDAR system is to significantly enhance the speed, accuracy, and completeness of assessment of battle damaged aircraft.

1.1 Background

AFRL completed an ABDAR project in 2000 that featured prototype software that demonstrated how software and digitized technical data could increase speed and accuracy of the ABDR process. ABDR experts from AFMC/MSG also completed an ABDAR software tool in 2002, referred to in this document as the MSG mockup. AFMC/MSG and the Combat Logistics Support Squadrons (CLSS) have shown continued interest in this capability, but it is unknown what portions of the mockup software packages, if any can be used to produce an ABDAR system that meets minimum CLSS requirements and can be fielded quickly. AFRL/HESR would like to determine existing capability within the current prototype and mockup and derive estimated cost figures for development of an initial ABDAR system based on the minimum requirements. This initial production will be called ABDAR 2K3.

1.2 Scope

The objective of the ABDAR evaluation effort is to assess the AFRL and MSG software currently demonstrating the concept of automating the documentation and repair of aircraft battle damage. The evaluation will determine what existing software code and processes, if any, are economically and technically valuable in developing a new product that meets minimum CLSS requirements and can be fielded in under six months. In addition, this software will be evaluated to determine the level of effort required to modify them in order to meet the minimal functional requirements.

The evaluation requirements are:

- Develop a set of criteria to evaluate the MSG and AFRL ABDAR software. The criteria will include as a minimum, functionality, data accessibility (digital TOs, Minimum Equipment System Listings- MESLs, etc.), expandability, design quality, code quality, software licensing, sustainability, and ease of loading onto a system in the field. The government will approve the final criteria list.

- Identify the crucial elements of the AFRL prototype software. Develop a matrix of the given requirements and evaluate how useful each element of the software will be in the immediate development of an ABDAR program intended to be fielded. Recommend alternative solutions for each element and identify the advantages and disadvantages of each alternative. Consideration shall be given for cost, flexibility, open architecture, expandability, and familiarity as a minimum for each element and alternative.

- Identify the vital elements of the MSG mockup software. Evaluate how useful each element of the MSG software will be in rapid development of a fielded version of ABDAR software. Develop a matrix of the given requirements. Recommend alternative solutions for each element if appropriate, and identify the advantages and disadvantages of each alternative. Consideration shall be given for cost, flexibility, open architecture, expandability, and familiarity as a minimum for each element and alternative.

- Compare and contrast elements of the MSG ABDAR mockup and the AFRL ABDAR prototype, and recommend the optimum solution to produce a rapid prototype that can be fielded in six months or less that meets the ABDAR minimum requirements. Provide an estimate of what it would cost to field a prototype within given constraints.

2. References

Title	Revision Date
System Specification for the Aircraft Battle Damage Assessment and Repair (ABDAR) System	5 February 1997
Aircraft Battle Damage Assessment and Repair (ABDAR) Technology - Updated Systems/Subsystem Specification (SSS)	10 March 2000
ABDAR Kick-Off Meeting Minutes	5 June 2003
Barbacci, Mario; Klein, Mark H.; Longstaff, Thomas H. & Weinstock, Charles B. Quality Attributes (CMU/SEI-95-TR-021). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1995.	
Boehm, Barry W.; Brown, John R.; Kaspar, Hans; Lipow, Myron; MacLeod, Gordon J. & Merritt, Michael J. Characteristics of Software Quality. New York, NY: North-Holland Publishing Company, 1978.	
Deutsch, Michael S. & Willis, Ronald R. Software Quality Engineering: A Total Technical and Management Approach. Englewood Cliffs, NJ: Prentice-Hall, 1988.	
Evans, Michael W. & Marciniak, John. Software Quality Assurance and Management. New York, NY: John Wiley & Sons, Inc., 1987.	
Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990.	
Software Engineering Institute [online]. Available WWW <URL: http://www.sei.cmu.edu> (2003).	
Webster's Ninth New Collegiate Dictionary. Springfield, MA: Merriam-Webster Inc., 1987.	

3. Requirements Traceability Matrix

For the purposes of cost estimation the requirements from the ABDAR System/Subsystem Specification (SSS) were broken up into five packages. The Requirements Traceability Matrix (see Appendix H) details the relationship between the five packages and the requirements.

4. Software Estimation Methodology

4.1 Background

Effective software estimation is one of the most difficult software development activities and one of the most important. Underestimating a project will lead to under staffing it, under scoping the quality assurance effort, and setting too short a schedule. That in turn can lead to staff burnout, low quality, loss of credibility as deadlines are missed, and ultimately to an inefficient development effort that takes longer than nominal. Overestimating a project can be almost as bad: Parkinson's Law that work expands to fill available time comes into play, which means that the project will take as long as estimated even if the project was overestimated. An accurate estimate is a critical part of the foundation of an efficient software project.

Software engineering studies have found that software project estimates created with the assistance of automated estimation software are more accurate than estimates created by manual methods. Automated estimation tools ultimately allow software projects to be delivered at lower cost than manual methods do.

The process for the selection of the estimation tool to be utilized led GRACAR to the Software Technology Support Center (STSC) at Hill AFB, UT. The STSC has conducted extensive research in the process of software estimation. The June 2002 edition of Crosstalk magazine was devoted to this subject. In this edition, the STSC identified two software tools available at no cost that provide sound estimation results. These tools were Cocomo 2.0 and Construx Estimate. Because of its ease of use, GRACAR selected Construx Estimate to utilize for this effort.

4.2 Calibration

Construx Estimate uses three distinct calibration approaches:

- Calibration by project type, which uses industry wide productivity data.
- Calibration by productivity drivers, which uses industry wide productivity data in conjunction with productivity adjustment factors.
- Calibration by historical data from the organization whose project is being estimated, which calibrates the estimation model based on historical data provided by the estimator.

4.2.1 Calibration by Project Type

Using Project Type is the least accurate means of calibrating an estimate. It is appropriate for creating a ballpark estimate before many detailed characteristics of the project can be known. In some circumstances it can produce accurate results.

- Select the project type.
- Select the project subtype.

The combination of project type and subtype can produce accurate estimates if the combination is selected carefully. The project type that the project being estimated most strongly resembles should be selected as the main project type. If the project has secondary characteristics that resemble one of the other project types, that type should be selected as the project subtype.

4.2.2 Calibration by Productivity Drivers

Productivity drivers are medium-accuracy means of calibrating an estimate. Among the productivity drivers utilized in the estimation process include the following:

- Complexity - Very simple; simple; average; complex; etc. We assigned this effort as average complexity except for Package 5 which received a complex score.
- Project Phase – Due to the work accomplished in the MSG mockup and the AFRL ABDAR software, we determined that the requirements definition phase has been completed.
- Personnel Experience – Analyst capability; programmer capability; experience in the area; experience with the platform; experience with the language and tools. We assigned the personnel as having three years experience with the area and the platform.
- Software Tools – Edit, code, debug; basic tools; mature tools; etc. It is assumed that the developer will be using mature tools for the ABDAR project.
- Application Familiarity – Unprecedented; somewhat unprecedented; familiar; etc. As in the complexity factor, the familiarity factor is very familiar except for Package 5 which received an unprecedented.
- Stability – Major change every year, minor every month; major change every 6 months, minor every month. We were not sure here so we utilized the major change every year with a minor every month.
- Documentation Requirements – Many life cycles need not be documented; documentation appropriate to life cycle; excessive life cycle documentation; etc. We utilized the documentation appropriate to the life cycle phase factor.

If you do not make a choice for an attribute, Construx Estimate will use the default value for that attribute. The more attributes you can describe, the more accurate your estimate will be.

The cost driver descriptions in this dialog box are based on software-industry-wide cost drivers. Select values that compare the project you are estimating to projects of all types.

4.2.3 Historical Data

Historical comparison data are the most accurate means of calibrating an estimate. Construx Estimate uses data from historical projects you select to estimate the productivity level of the estimate.

Add projects from the list at the top of the dialog box to the list at the bottom.

- First select the row you would like to add from the top list
- Press the Add to List button to add projects to bottom list.

Here are some guidelines for selecting projects that will give you the most accurate estimate:

- Choose projects that used the same people that will work on the project you're estimating. For small projects (less than 5 people), this is the most significant consideration. For larger projects, it is still a significant factor.
- Choose projects that are technically similar to the project you are estimating. Choose projects that resemble your project in size, in project type, and programming language.
- Try to choose at least three similar projects. Don't force fit a project that doesn't resemble your project, but, statistically, three projects provide a much better basis for the estimate than one or two.

GRACAR has selected the use of a combination of Calibration by Project Type and Calibration by Productivity Drivers. We were not able to obtain historical data that would be representative of similar projects.

4.3 Goal Seeking

Construx Estimate uses sophisticated goal-seeking algorithms that can find an optimum staffing or schedule based on constraints you enter for effort, schedule, cost, and peak manpower.

You can also choose to enter relative priorities for effort, schedule, cost, and peak manpower. Construx Estimate will compute a solution that balances these priorities to the maximum extent possible. You can also use Estimate's "What If" capability to do your own goal seeking and see the effect of different estimation inputs on the estimation outputs.

4.4 Probability of Success

Construx Estimate contains a sophisticated statistical simulation module that predicts the likely outcomes of a project based on the calibration data and project data you have entered. For any given cost, schedule, or level of effort, you can see what the probability of achieving that outcome. For example, if Construx Estimate recommends a 12-month schedule but your plans call for a 9-month schedule, you can see whether your chances of completing the project within 9 months are 75 percent or 5 percent.

4.5 Estimation Algorithms

Construx Estimate makes use of three mature estimation approaches:

4.5.1 SLIM

SLIM was developed by Lawrence H. Putnam in the early 1970s and first offered as a commercial product in 1978. This methodology has been continuously refined since its initial offering and is fully described in a book Putnam co-authored with Ware Myers, Measures for Excellence (Yourdon Press, 1992).

The SLIM methodology is based on the insight that efficiently run software projects follow well-defined patterns that can be modeled with a set of exponential equations. These equations form the backbone of Construx Estimate's approach to creating cost, schedule, peak staffing, and defect estimates.

4.5.2 Cocomo 2.0

Cocomo 2.0 is a continuation of the work begun by Barry W. Boehm in the 1970s and described in his 1981 book, Software Engineering Economics (Prentice-Hall). Since 1981, additional work has been done to refine the Cocomo 2.0 model and adapt it to projects other than the U.S. Department of Defense projects for which it was originally developed. At present, the model has been extended into Cocomo 2.0, which allows estimates to be created for virtually any kind of project by specifying a set of cost drivers. Construx Estimate uses the Cocomo 2.0 model as a supplement to the SLIM model when estimates are calibrated using cost drivers. A productivity baseline is established using the project type settings; the productivity factor is then adjusted using the computed Cocomo 2.0 productivity. Construx Estimate uses Cocomo 2.0 data and algorithms from Cocomo 2.0 Model Definition Manual, version 1.4.

4.5.3 Monte Carlo Simulation

Construx Estimate uses Monte Carlo simulations to model complex interactions in the face of uncertain estimating assumptions. Construx Estimate simulates hundreds or thousands of possible outcomes of the project being estimated based on size, productivity, current project phase, and other parameters entered by the estimator. It then estimates the likelihood of various project outcomes and assigns risk levels to different planning options. In complex situations that involve a lot of uncertainty, the methodology allows Construx Estimate to create meaningful estimates that would otherwise be impossible to model.

4.6 Construx Estimate Refinement

Construx Estimate allows you to create rough estimates early in a project and to refine those estimates as the project progresses. Early in the project, you can create more accurate estimates than those derived from purely non-automated techniques. Later in the project, as you gain more insight into project scope and other characteristics, you can refine your estimate and improve its accuracy.

4.7 Cost Estimation

As part of its project reporting, Construx Estimate provides an estimated cost for the development. To develop the estimate the system requires the input of labor rates for three categories. These categories are project management, application development, and quality assurance. GRACAR used the rates found in the table below for the

ABDAR project. Actual cost will be dependent upon many factors including the contractor selected for development. The cost estimations contained in this report uses the following hourly labor rates:

Labor Category	Labor Rate
Project Management	$ 80.00
Application Development	$ 70.00
Quality Assurance	$ 65.00

Table 4-1: Estimated Labor Rates

4.8 Basis of Estimation

Construx Estimate is able to utilize either Lines of Code or Classes as the input to be utilized for its computation. When classes are used as a basis of the estimation, the system will then derive expected lines of code upon completion of the process. For this report, classes were utilized as the basis for the evaluation. The table below represents the estimated number of classes:

Package	Classes
1	33
2	45
3	391
4	213
5	83

Table 4-2: Class Estimate

5. System Evaluation

5.1 Background

As Identified in the Statement of Work, the purpose of this project was to evaluate the ABDAR demonstration prototype developed by AFRL and the MSG mockup to assess which was more complete and could be brought to the field as an actual product, and what the effort would be to accomplish this task.

The *Software Quality Criteria* (See Appendix F) identifies the definitions that were used in making the assessment of the two pieces of software. These definition titles were transferred to the ABDAR Implementation Criteria matrix in order to weight each demonstration against these definitions. This section describes the process of applying the Software Quality Criteria definitions to the two demonstrations to determine the weight given each for each individual category, and ultimately the final grade.

5.2 Approach

5.2.1 Software Quality Criteria

Sixteen software quality criteria were used during the evaluation. See Appendix F for a definition of the criteria. The sixteen criteria are:

1. Effectiveness
2. Responsiveness
3. Correctness
4. Verifiability
5. Usability
6. Fidelity
7. Dependability
8. Efficiency/Resource Utilization
9. Maintainability
10. Understandability
11. Interoperability
12. Portability
13. Scalability
14. Reusability
15. Cost of Ownership
16. Productivity

These sixteen criteria are organized into four groups.

1. Functional Analysis
2. Human Factors
3. Technical Design
4. Implementation

5.2.2 ABDAR Implementation Criteria matrix

The matrix that is represented in the ABDAR Implementation Criteria (see Appendix G) is formatted in the following manner:

The left-hand vertical column is divided into the four major groups defined in the Software Quality Criteria document:

1. Functional Analysis
2. Human Factors
3. Technical Design
4. Implementation

Each of the four groups is further divided by the software quality criteria that have been assigned to each of these groups. Each of these criteria was then sub-divided to show the score values for the grades of A, B, C, D, and F.

Letter Grade	Score
A	100.0
B	77.5
C	55.0
D	32.5
F	10.0

Table 5-1: Evaluation Grade Values

There are three major columns along the horizontal axis of the matrix. These columns are as follows:

- Weight (The column weightings)
- AFRL
- MSG

Each of these columns was then weighted and broken down into the following five areas:

1. Do not need to have
2. Nice to have
3. Somewhat necessary
4. Very desirable
5. Has to be present

Each of the sixteen software quality criteria used was given a weighting factor that affects the net score of the demonstration software for that category.

Criteria	Necessity	Weight Factor
Effectiveness	Has to be present	1.000
Responsiveness	Has to be present	1.000
Correctness	Has to be present	1.000
Verifiability	Has to be present	1.000
Usability	Very desirable	0.775
Fidelity	Has to be present	1.000
Dependability	Has to be present	1.000
Efficiency/Resource Utilization	Somewhat necessary	0.550
Maintainability	Has to be present	1.000
Understandability	Has to be present	1.000
Interoperability	Somewhat necessary	0.550
Portability	Do not need to have	0.100
Scalability	Somewhat necessary	0.550
Reusability	Somewhat necessary	0.550
Cost of Ownership	Somewhat necessary	0.550
Productivity	Has to be present	1.000

Table 5-2: Software Quality Criteria Weighting

Each demonstration was then evaluated on the software quality criteria and given a letter grade. A weighted score was placed where the horizontal and vertical axis intersected based on the necessity of the definition in the software (horizontal axis) against how well it achieved the definition (vertical axis). The weighted score is achieved by multiplying the score value of the letter grade by the weight factor of the necessity.

5.2.3 Calculation of Final Grades

All of the weighted scores from the software quality criteria were then averaged (using the sum of the necessity weights) to arrive at an average weighted score ranging from 0.0 to 100.0

5.3 Software Evaluation

5.3.1 MSG

Based on several conversations and meetings that GRACAR had with Air Force Research Laboratory personnel from HESR and LGRC, it was determined that the MSG mockup represented the graphical user interface preferred by the end user. It should be noted that GRACAR was only given a conceptual mockup done as a Flash presentation.

Flash is a presentation package that is produced by Macromedia. While the MSG presentation gives the appearance of a prototype, it is just a presentation, there is no code associated with it, other than the scripting language that is proprietary to the Macromedia Flash package.

There is nothing that can be brought forward and developed into an actual application other than the look and feel. Because of this, the MSG mockup received low marks in all categories. The highest mark for the MSG mockup was in Usability because of its look and feel.

Report Card		
Major Group	**Sub-Definitions**	**Score**
Functional Analysis	Effectiveness	10.0
	Responsiveness	10.0
	Correctness	10.0
	Verifiability	10.0
Human Factors	Usability	25.2
	Fidelity	10.0
Technical Design	Dependability	10.0
	Efficiency/Resource Utilization	5.5
	Maintainability	10.0
	Understandability	10.0
Implementation	Interoperability	5.5
	Portability	1.0
	Scalability	5.5
	Reusability	5.5
	Cost of Ownership	5.5
	Productivity	10.0

Table 5-3: MSG Mockup Report Card

On a scale of 0 to 100, the MSG software received a final score of **11.4**. This is an average of all the scores.

5.3.2 AFRL

While the AFRL prototype was indeed a true prototype, there were several factors that caused it to receive low marks.

1) The Java server (Tanga) that the AFRL prototype software uses is no longer produced. The Tanga server was produced by BEA Inc. It has been replaced with the WebLogic server. The current WebLogic server specifications state that it is backward compatible to the last version the Tanga server, however the Tanga server that the AFRL prototype uses is more than two versions prior to the last version of Tanga that was produced. In addition, several of the key WebLogic Java Class names have changed and they kept the older Tanga Class names as well. Because of this, it was determined that the effort to bring the prototype forward to the current WebLogic server, would require comparing all of the Tanga Java Server classes with the current WebLogic Java Server classes for compatibility. This would amount to an effort that would rival writing the application from scratch.

2) There were several third-party Java Class Libraries that were used in writing the prototype that are no longer available or supported. These classes provide support that has been incorporated into the standard Java Class libraries from Sun, so the third party libraries were no longer necessary.

3) The AFRL prototype displayed technical documents in PDF format. However, this was external to the application and not imbedded in the application. Even though the AFRL prototype was able to interact with the Adobe Reader, it made for a very disjointed user experience.

4) In addition to and because of the drawbacks listed above, the steps that were required to start the AFRL prototype were cumbersome and unstable. The effort to bring the AFRL software forward to a usable, stable, user friendly application would be far greater than the effort to write the application from scratch.

For these reasons the AFRL prototype, while receiving higher marks than that of the MSG presentation, received very low marks.

Report Card		
Major Group	**Sub-Definitions**	**Weighted Score**
Functional Analysis	Effectiveness	32.5
	Responsiveness	10.0
	Correctness	32.5
	Verifiability	10.0
Human Factors	Usability	7.8
	Fidelity	32.5
Technical Design	Dependability	10.0
	Efficiency/Resource Utilization	30.3
	Maintainability	10.0
	Understandability	10.0
Implementation	Interoperability	17.9
	Portability	3.3
	Scalability	5.5
	Reusability	17.9
	Cost of Ownership	5.5
	Productivity	32.5

Table 5-4: AFRL Prototype Report Card

On a scale of 0 to 100, the AFRL software received a final score of **21.2**. This is an average of all the scores.

5.4 Evaluation Conclusion

The evaluations of both the MSG mockup and the AFRL prototype were reviewed in the context of moving forward with fielding a minimum ABDAR capability as defined by during meetings with users. Both the tools evaluated provided significant value and insight to the user interface and functional process flow being automated. For example, the ABDAR prototype software represent years of requirements analysis. That analysis can be carried forward to a future development, and in the process, provide high value in terms of cost saving associated with requirements analysis and design phases associated with any development. The MSG mockup provides less in terms of functional requirements analysis, but does render valuable feedback from the user community in terms of the look and feel desired from the current technological baseline.

In light of the technical evaluations of the MSG and AFRL products, we determined that neither could serve as the technical basis for the development of the ABDAR production system. This determination was based on a number of factors that came to light during the evaluation and are highlighted below.

- The MSG mockup was created in a week using some HTML and a flash demo. While there is value in the user interface being demonstrated, from a technical development perspective, the mockup will offer less than a week of development savings, based solely on the time it took to create the mockup. The value of moving forward with any code generated to produce the mockup would be negligible in the context of an entire development effort.

- The AFRL prototype was developed several years ago and that Java has matured greatly as a development language since the AFRL prototype was developed. In terms of moving forward with the code in the AFRL prototype, the costs of reverse engineering the current code and bringing it in line with current Java standards and development techniques would exceed the cost of a new start to develop the functional requirements represented in the AFRL product.

- The AFRL product was specifically designed for use with the F-15 weapon system. Re-use of the code used in the AFRL product would render another product designed for use only on the F-15 and one that could not be extended for use on other weapon systems. We viewed this as unacceptable in response to the users requirements.

As a result of the aforementioned factors, GRACAR recommends that the ABDAR application be written from scratch, using requirements analysis and artifacts from both the MSG and AFRL efforts to greatly enhance the chances for success of a new development project. Doing so will provide substantial reductions in the time and development costs associated a new effort.

6. Proposed Solution

6.1 Discussion

As mentioned in Section 5.4, a recommendation has been made that neither the MSG nor the AFRL packages are acceptable as the basis for the ABDAR system. Based on GRACAR's recommendation, the proposed solution will consist of options for the development of a new system that consistent with the current USAF architectures and offers the best opportunities for future expansion.

6.2 Solution

The ABDAR solution will be presented as a series of five (5) capabilities packages. The package will consist of a logical grouping of functional requirements as outlined in the ABDAR System/Subsystem Specification (SSS). Packages start with the most basic capabilities concluding with an integrated system designed to function on an internal network.

6.2.1 Requirements

As with the other proposed solutions, we have several functional and technical requirements. These requirements were derived from either the Statement of Work (SOW) or subsequent project meetings were:

- Minimum level of functionality shall be an automated AFTO Form 97 with hooks to fill in the form
- Highest level of functionality shall a completely integrated system
- At a minimum, PDF shall be the lowest level of electronic technical data
- Export function is required to facilitate the passage of data between ABDAR and other maintenance systems (CAMS, G081, etc.)
- Required architecture is defined as a standalone application. When a server is employed, the server must be GCSS-AF compliant
- WEB application is not acceptable
- If possible, the first iteration of the software should be deployable within a six (6) month period

6.2.2 ABDAR Application Packages

The following sections provide a short summary of the functionality to be derived from each individual package. The Requirements Traceability Matrix (RTM) found in Appendix H will show the detailed functional requirements found in each package.

6.2.2.1 Package 1 – Initial Software Capability

This package is designed to meet the stated minimum ABDAR requirements. It will consist of providing the capability to interactively complete the AFTO Form 97 as well as displaying the electronic TO data. This package will provide the foundation on which all subsequent packages will be built upon.

6.2.2.2 Package 2 – Remaining Minimum Capability

This package consists of all the remaining functional capabilities that have been previously identified as minimum requirements in a standalone environment. This package, along with Package 1, comprises the desired system in a standalone environment.

6.2.2.3 Package 3 – Additional Requested Standalone Capabilities

This package finalized all the remaining requirements for ABDAR that will operate in a standalone environment.

6.2.2.4 Package 4 –Network Required Capabilities

The final package consists of all functional requirements that will depend on the availability of a network in order to function. These requirements greatly enhance the capabilities of the system to augment the operational infrastructure.

6.2.2.5 Package 5 – Integrated Technical Order (TO) Database Excursion

This package will provide the enhanced capability to both access and retrieve data from their applicable TOs. The package will perform the necessary actions to create the database as well as any required maintenance or updates to the data. It should be pointed out that this package is unique in that the effort to develop this package may be recursive. That is a new version of this package ***may*** be required for each types of TO or for TOs that are substantially different from the standard documents.

6.3 Size/Cost Estimations

As mentioned in Section 4.1, GRACAR has selected Construx Estimate as the estimation tool to be employed on this project. The basis for the estimation was an analysis of the functionality and the number of classes required to satisfy the requirement. Each package was evaluated individually in order to provide the customer with the options of choosing the desired solution.

The Construx Estimate tool computed both Nominal and Optimum solutions. The Nominal plan represents a 50/50 estimate – the estimate for which there is both a 50 percent chance of overrunning and a 50 percent chance of under running the estimate. The Optimum plan utilizes the project productivity drivers and meets the project entire set of constraints and priorities to the maximum extent possible. A complete copy of the estimation reports can be found in Appendices A-E to this report. The following represents a summary of the estimation results for each package:

6.3.1 Package 1 – 33 Classes

6.3.1.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	17
Schedule (calendar months)	8.4
Cost	$178,581
Peak Staff (people)	2.7
Average Staff (people)	2.0

Table 6-1: Package 1 Nominal Plan

6.3.1.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	11
Schedule (calendar months)	9.3
Cost	$115,327
Peak Staff (people)	1.6
Average Staff (people)	1.2

Table 6-2: Package 1 Optimum Plan

6.3.2 Package 2 – 45 Classes

6.3.2.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	25
Schedule (calendar months)	9.8
Cost	$277,914
Peak Staff (people)	3.5
Average Staff (people)	2.6

Table 6-3: Package 2 Nominal Plan

6.3.2.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	15
Schedule (calendar months)	14.2
Cost	$164,446
Peak Staff (people)	1.8
Average Staff (people)	1.3

Table 6-4: Package 2 Optimum Plan

6.3.3 Package 3 – 391 Classes

6.3.3.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	459
Schedule (calendar months)	25.8
Cost	$5,075,329
Peak Staff (people)	27.7
Average Staff (people)	17.8

Table 6-5: Package 3 Nominal Plan

6.3.3.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	272
Schedule (calendar months)	29.4
Cost	$3,003,153
Peak Staff (people)	14.4
Average Staff (people)	9.2

Table 6-6: Package 3 Optimum Plan

6.3.4 Package 4 – 213 Classes

6.3.4.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	154
Schedule (calendar months)	17.1
Cost	$1,640,431
Peak Staff (people)	14.0
Average Staff (people)	9.0

Table 6-7: Package 4 Nominal Plan

6.3.4.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	91
Schedule (calendar months)	19.5
Cost	$970,669
Peak Staff (people)	7.3
Average Staff (people)	4.7

Table 6-8: Package 4 Optimum Plan

6.3.5 Package 5 - 83 Classes

6.3.5.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	57
Schedule (calendar months)	12.94
Cost	$631,456
Peak Staff (people)	6.3
Average Staff (people)	4.4

Table 6-9: Package 5 Nominal Plan

6.3.5.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	34
Schedule (calendar months)	14.7
Cost	$373,642
Peak Staff (people)	3.3
Average Staff (people)	2.3

Table 6-10: Package 5 Optimum Plan

6.4 Estimation Quality

Estimates vary in the quality of the assumptions used to create them. Some of these characteristics can be evaluated programmatically. This report rates the quality of the estimate on a 5 point verbal scale:

- Excellent
- Very Good
- Good
- Fair
- Poor

6.4.1 Calibration Evaluation

GRACAR has selected the use of a combination of Calibration by Project Type and Calibration by Productivity Drivers. We were not able to obtain historical data that would be representative of similar projects.

6.4.2 Scope Evaluation

This estimate's type of scope estimate: Basic Size (Classes/Modules)

6.4.3 Phase Evaluation

Estimates created later in a project are more accurate than estimates created early in the project. Even the best estimates cannot be very accurate if they are created at a point in the project when comparatively little is known about the software to be built.

Current Project Phase: General Requirements Complete

6.4.4 Suitability Evaluation

The Construx Estimate tool works effectively when at least two of the following conditions are met:

- Estimated size is greater than or equal to 5000 lines of code
- Nominal development is expected to be at least 6 months
- Nominal effort is expected to be at least 18 staff-months
- Nominal peak staffing is at least 3 people

This project uses Project type (from industry data) calibration and four of these conditions have been met.

The estimation quality for the ABDAR packages is as follows:

Evaluation Factor	Rating
Calibration	Good
Scope	Good
Phase	Fair
Suitability	Excellent
Overall	**Good**

Table 6-11: Estimation Quality

6.5 Reusability

As mentioned in Section 5.4 of this report, both the AFRL and the MSG software packages have been found unsuitable for use and therefore the estimates provided above are based on a completely new development. While the demonstrations are unsuitable for use, the artifacts from the previous development efforts may be valuable in the future. Unfortunately it is extremely difficult to quantify the reuse benefits.

The development of the functional requirements derived from the previous effort will prove invaluable to any future efforts. Additionally, features such as the user interface, wizards, etc can certainly became the basis for the system in the future.

6.6 Risks

While it might be unusual to be identifying risks during an evaluation phase, there is one aspect to the ABDAR program that certainly warrants examination. The feasibility of the system is dependent upon the availability of TOs in an electronic format. Moreover, it would advantageous to the program if the documents were also available in an XML format.

Our research, while not completely conclusive, indicates that the distribution of TOs in an electronic format is currently less than 10%. It is difficult to ascertain whether the required TOs will be available in the correct format when required.

Additionally, any commonality of format will reduce the cost significantly. A portion of the software to be developed in Package 2 will be dependent upon the physical format of the TO. In a worse case scenario it may be necessary to develop a Package 5 solution that is unique to the weapon system.

7. Certification and Accreditation (C&A)

7.1 Background

The C&A requirements for ABDAR will vary greatly dependent upon the solution and architecture selected. The selection of a network solution will greatly increase the C&A tasks. The utilization of a true standalone solution could considerably reduce those same tasks.

7.2 Command, Control, Communications, Computer and Intelligence Support Plan (C4ISP)

Paragraph 2.1 of the Interim Guidance for Developing, Processing and Approving C4ISPs, Networthiness and Systems Certification states that it applies to all "programs/systems that connect in any way with the Air Force communications and information infrastructure". The guidance further states that if the Program Manger (PM) does not feel that a C4ISP is required, the PM must conduct a self-assessment to determine the C4ISR impacts to their system. It is our judgment that in the standalone mode an ABDAR will not touch the communications infrastructure and as such a C4ISP will not be required.

If it determined that ABDAR will employ a network environment this situation will need to reexamined. At that point the system will utilize the infrastructure. Assuming that ABDAR will not be MAJCOM unique system, a full C4ISP will be required. Estimated development time for the document will be approximately 200 man-hours. The coordination cycle for a C4ISP can be as long as seven (7) months.

7.3 System Security Authorization Agreement (SSAA)

According to the DoD Information Technology Security Certification and Accreditation Process (DITSCAP) all systems require an SSAA. Again, systems that operate in a standalone manner fall into a gray area for compliance with the regulation. Waiver to this regulation should be investigated. If necessary, an estimated 100 man-hours will be required for the preparation of the SSAA.

7.4 Certificate of Networthiness (CON)

As with the items described above, the CON will only be necessary if the system operates in a network environment.

8. Conclusion

GRACAR's summary of findings focused on three major areas.

- Analysis of Existing AFRL/MSG software
- Estimation of Effort to Develop a New System
- Risks

8.1 Analysis of Existing AFRL/MSG Software

The evaluations of both the MSG mockup and the AFRL prototype were reviewed in the context of moving forward with fielding a minimum ABDAR capability as defined by during meetings with users. Both the tools evaluated provided significant value and insight to the user interface and functional process flow being automated. For example, the ABDAR prototype software represent years of requirements analysis. That analysis can be carried forward to a future development, and in the process, provide high value in terms of cost saving associated with requirements analysis and design phases associated with any development. The MSG mockup provides less in terms of functional requirements analysis, but does render valuable feedback from the user community in terms of the look and feel desired from the current technological baseline. Reference Section 5 of this report for more detailed analysis of both the AFRL and MSG software packages. Both packages received a score that is deemed unsuitable to be utilized as the basis for the production system. Both efforts have resulted in the production of reusable artifacts that will enhance the chances of success of any future project. With this in consideration we still feel that it would not be cost effective to utilize the actual code as the basis for production ABDAR system.

8.2 Estimation of Effort to Develop a New System

To arrive at an estimation of the resources required to develop a new system, GRACAR utilized a commercial package called Construx Estimate. The basis of the estimation was our analysis of the number of classes that would be required to fulfill the stated user requirements. The Analysis determined that a total of five (5) development packages would be required. The packages are as follows:

Package	Description	Man-Months	Man-Months (Cumulative)	Cost	Cost (Cumulative)
1	Initial Software Capability	11	11	$115,327	$115,327
2	Remaining Minimum Capability	15	26	$164,446 Requires Package 1	$279,773
3	Additional Requested Standalone Capabilities	272	298	$3,003,153 Requires Packages 2 & 1	$3,282,926
4	Network Required Capabilities	91	389	$970,669 Requires Packages 3, 2, & 1	$4,253,595
5	Integrated Technical Order (TO) Database Excursion	34	423	$373,642 Standalone	$4,627,237

Table 8-1: Software Package Estimates

8.3 Recommendation

It is GRACAR's recommendation that ABDAR be a new development and we have outlined the following steps for consideration:

Step 1 - Develop Packages 1 & 2. Track the progress of digital TOs closely and attempt to draw upon their work. If this not possible, consider Package 5 in parallel with 1 and 2
Step 2 - Revalidate Functional Requirements of Packages 3 & 4
Step 3 - Develop Package 3
Step 4 - Develop Package 4 if warranted as a result of the revalidation effort in step 2

We feel that the above represents the safest and most cost effective path for the future of the ABDAR development.

9. Notes

9.1 Acronyms

A/C	Aircraft
ABD	Aircraft Battle Damage
ABDAR	Aircraft Battle Damage Assessment and Repair
ABDR	Aircraft Battle Damage Repair
ACC	Air Combat Command
AFMC	Air Force Materiel Command
AFOSH	Air Force Occupational Safety and Health
AFSC	Air Force Specialty Code
AFSOC	Air Force Special Operations Command
AFTO	Air Force Technical Order
ALC	Air Logistics Center
AMC	Air Mobility Command
ARED	ABDAR Requirements Database
ATOS	Automated Technical Order System
BDR	Battle Damage Repair
BIT	Built-In Test
C4ISP	Command, Control, Communications, Computer and Intelligence Support Plan
C&A	Certification and Accreditation
CALS	Continuous Acquisition of Life Cycle Support
CAMS	Core Automated Maintenance System
CD	Chemical Defense
CEMS	Comprehensive Engine Management System
CFRS	Computerized Fault Reporting System
CLSS	Combat Logistics Support Squadrons
CON	Certificate of Networthiness
DII-COE	Defense Information Infrastructure-Common Operating Environment
DITSCAP	DoD Information Technology Security Certification and Accreditation Process
DTC	Data Transfer Cartridge
EOD	Explosive Ordnance Disposal
EPA	Environmental Protection Agency
ETIC	Estimated Time in Commission
ETTC	Estimated Time to Complete
ETTR	Estimated Time(s) To Repair
FMC	Fully Mission Capable
FOM	Facilitate Other Maintenance
GSE	Ground Support Equipment
GUI	Graphical User Interface
HAZMAT	Hazardous Materials
HCI	Human Computer Interface
IAW	In Accordance With

IETM	Integrated Electronic Technical Manuals
IMDS	Integrated Maintenance Data System
IMIS	Integrated Maintenance Information System
IPB	Illustrated Parts Breakdown
IPI	In-Process Inspections
JCN	Job Control Numbers
LAN	Local Area Network
LRU	Line Replace Units
MACC	Maintenance Aircraft Coordination Center
MAJCOM	Major Command
MC	Mission Capable (includes both FMC and PMC)
MESL	Mission Essential Subsystems List
MFL	Maintenance Fault List
MOC	Maintenance Operations Center
MOI	Maintenance Operating Instructions
MOPP	Mission Oriented Protective Posture
MSDS	Material Safety Data Sheet
NDI	Nondestructive Inspection
NMC	Not Mission Capable
NMCB	Not Mission Capable - Both
NMCM	Not Mission Capable - Maintenance
NMCS	Not Mission Capable - Supply
OEM	Original Equipment Manufacturer
OFP	Operational Flight Program
OIC	Officer In Charge
OS	Operating System
PMA	Portable Maintenance Aid
PMC	Partially Mission Capable
PMCB	Partially Mission Capable - Both
PMCM	Partially Mission Capable - Maintenance
PMCS	Partially Mission Capable - Supply
POC	Point(s) Of Contact
QPA	Quantity Per Application
REMIS	Reliability and Maintainability Information System
SA/BC	Self Aid/Buddy Care
SBSS	Standard Base Supply System
SMR	Source, Maintenance, and Recoverability
SOP	Standard Operating Procedures
SOW	Statement of Work
SPD	System Program Director
SPM	System Program Manager
SPO	System Program Office
SRD	Standard Reporting Designator
SSAA	System Security Authorization Agreement
SSS	System Specification
SURVIAC	Survivability Vulnerability Information Analysis Center

TCTO	Time Compliance Technical Orders
TICARRS	Tactical Interim CAMS and REMIS Reporting System
TO	Technical Order
TPFDL	Time Phased Force Deployment List(ing)
TRAP	Tanks, Ranks, Adapters and Pylons
USAF	United States Air Force
UXO	Unexploded Ordnance
WAN	Wide Area Network
WSMIS	Weapon System Management Information System
WUC	Work Unit Code

10. Appendix A: ABDAR Software Estimate – Package 1

10.1 Estimate Summary

This document provides the estimate for the minimal functional capability as described in the ABDAR documentation. This estimation provides for the preparation of an AFTO 97 as well as the method for the import, display, and initial retrieval of USAF Technical Orders (TOs) utilizing source documents of a PDF format.

10.1.1 Nominal Plan

Current Project Phase: General Requirements Complete

Management Metric	Expected Value (50% Probability)	Standard Deviation	Standard Deviation as Percentage
Lines of Code	31,983	1,504	+/- 5%
Man-Months	17	15	+/- 88%
Schedule (calendar months)	8.4	1.7	+/- 20%
Cost	$178,581	$156,830	+/- 88%
Peak Staff (People)	2.7	1.5	+/- 55%
Average Staff (People)	2.0	1.8	+/- 88%
Overall Estimate Quality	Good		

Table 10-1: Package 1 Nominal Estimate

This estimate is the 50/50 estimate – the estimate for which there is both a 50 percent chance of overrunning and a 50 percent chance of under running the estimate. This is also known as the nominal estimate. This estimate is for the "main build" phase of a project, the time from detailed requirements specification complete to software acceptance. Earlier phases of a project are not estimated here.

10.1.2 Optimum Plan

Management Metric	Optimum Planning Value
Effort (Staff Months)	11
Schedule (calendar months)	9.3
Cost	$115,327
Peak Staff (People)	1.6
Average Staff (People)	1.2

Table 10-2: Package 1 Optimum Estimate

These planning values meet the projects entire set of constraints and priorities to the maximum extent possible.

10.2 Estimate Quality

10.2.1 Summary of Estimate Quality

Estimates vary in the quality of the assumptions used to create them. Some of these characteristics can be evaluated programmatically. This report rates the quality of the estimate on a 5 point verbal scale:

- Excellent
- Very Good
- Good
- Fair
- Poor

Overall quality of this estimate: Good

10.2.2 Calibration Evaluation

Estimates calibrated with three or more historical projects are most accurate. Estimates calibrated with one or two historical projects, cost drivers, or project types are less accurate. This estimate has been calibrated using project type.

Calibration Quality: Good

10.2.3 Scope Evaluation

Scope estimates created with fine-granularity units such as lines of code include less imprecision than estimates created with large-granularity units such as classes and subsystems.

This estimate's type of scope estimate: Basic Size (Classes/Modules)

Scope Estimate Quality: Good

10.2.4 Phase Evaluation

Estimates created later in a project are accurate than estimates created early in the project. Even the best estimates cannot be very accurate if they are created at a point in the project when comparatively little is known about the software to be built.

Current Project Phase: General Requirements Complete

Estimate quality possible in this phase: Fair

10.2.5 Consistency Check

The table below provides a consistency check by comparing the current project estimate to results from other projects of similar sizes and types. The estimated project has a type of “Business Systems”.

Management Metric	Value	Assessment
Productivity (lines of code per staff-month)	2,951	Within Normal Range
Schedule (Months)	9.3	Within Normal Range
Effort (Staff-Months)	11	Within Normal Range
Average Staff (people)	1.2	Within Normal Range
Code Generation Rate (lines of code per month)	3,432	Within Normal Range

Table 10-3: Package 1 Consistency Check

10.2.6 Suitability Evaluation

The Estimate tool works effectively when at least two of the following conditions are met:

- Estimated size is greater than or equal to 5000 lines of code
- Nominal development is expected to be at least 6 months
- Nominal effort is expected to be at least 18 staff-months
- Nominal peak staffing is at least 3 people

When less than two of these criteria are met, the only way to achieve a reliable estimate is to use historical calibration. Even when historical calibration is used, some projects are too small to estimate reliably. This project uses Project type (from industry data) calibration and four of these conditions have been met.

Suitability of Construx Estimate to estimate this project: Excellent

10.3 Planning Options Overview

10.3.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	17
Schedule (calendar months)	8.4
Cost	$178,581
Peak Staff (people)	2.7
Average Staff (people)	2.0

Table 10-4: Package 1 Planning - Nominal

10.3.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	11
Schedule (calendar months)	9.3
Cost	$115,327
Peak Staff (people)	1.6
Average Staff (people)	1.2

Table 10-5: Package 1 Planning – Optimum

10.3.3 Shortest-Schedule Plan

Management Metric	Planning Value
Effort (staff months)	41
Schedule (calendar months)	6.7
Cost	$435,991
Peak Staff (people)	8.2
Average Staff (people)	6.1

Table 10-6: Package 1 Planning – Shortest Schedule

10.3.4 Least-Cost Plan

Management Metric	Planning Value
Effort (staff months)	6
Schedule (calendar months)	10.9
Cost	$62,526
Peak Staff (people)	0.7
Average Staff (people)	0.5

Table 10-7: Package 1 Planning-Least Cost

10.4 Priorities

Priorities can be used to determine the optimal project plan. The table below shows the priorities used to create this estimate.

Priority	Value
Schedule Priority	High Priority
Effort Priority	High Priority
Cost Priority	High Priority
Peak Staff Priority	Medium Priority

Table 10-8: Package 1 Priorities

10.5 Scope Probabilities

The table below contains scope estimates by probability. These scope estimates are expressed in lines of code. If the scope estimates were not originally expressed by the estimator in lines of code, they have been converted to lines of code. The scope estimates are based on parameters that have been entered by the estimator, including the following:

Scoping Method: Basic Size (Classes/Modules)

Project Phase: General Requirements Complete

Number of Simulations: 500

Probability (%)	Scope Will Be Less Than	Difference From Nominal
1.0	28,500	-11%
5.0	29,450	-8%
10.0	30,241	-5%
20.0	30,716	-4%
30.0	31,033	-3%
40.0	31,666	-1%
50.0	31,983	0%
60.0	32,300	1%
70.0	33,091	3%
80.0	33,408	4%
90.0	33,883	6%
95.0	34,516	8%
99.0	35,466	11%

Table 10-9: Package 1 Scope Probabilities

10.6 Effort Probabilities

The table below contains effort estimates by probability. The effort estimates are expressed in staff-months. They are based on parameters that have been entered by the estimator, including the following:

Calibration Method: Project Type

Project Phase: General Requirements Complete

Number of Simulations: 500

Probability (%)	Effort Will Be Less Than	Difference From Nominal
1.0	7	-65%
5.0	10	-53%
10.0	11	-48%
20.0	13	-36%
30.0	15	-27%
40.0	19	-12%
50.0	21	0%
60.0	25	16%
70.0	29	39%
80.0	37	75%
90.0	57	171%
95.0	114	440%
99.0	1,892	8,842%

Table 10-10: Package 1 Effort Probabilities

11. Appendix B: ABDAR Software Estimate – Package 2

11.1 Estimate Summary

This document provides the estimate of the remainder of what is considered the minimal functional capability as described in the ABDAR documentation.

11.1.1 Nominal Plan

Current Project Phase: General Requirements Complete

Management Metric	Expected Value (50% Probability)	Standard Deviation	Standard Deviation as Percentage
Lines of Code	39,108	1,164	+/- 3%
Man-Months	25	15	+/- 61%
Schedule (calendar months)	9.8	1.9	+/- 19%
Cost	$277,914	$170,654	+/- 61%
Peak Staff (People)	3.5	1.2	+/- 36%
Average Staff (People)	2.6	1.6	+/- 61%
Overall Estimate Quality	Good		

Table 11-1: Package 2 Nominal Estimate

This estimate is the 50/50 estimate – the estimate for which there is both a 50 percent chance of overrunning and a 50 percent chance of under running the estimate. This is also known as the nominal estimate. This estimate is for the "main build" phase of a project, the time from detailed requirements specification complete to software acceptance. Earlier phases of a project are not estimated here.

11.1.2 Optimum Plan

Management Metric	Optimum Planning Value
Effort (Staff Months)	15
Schedule (calendar months)	11.2
Cost	$164,446
Peak Staff (People)	1.8
Average Staff (People)	1.3

Table 11-2: Package 2 Optimum Estimate

These planning values meet the projects entire set of constraints and priorities to the maximum extent possible.

11.2 Estimate Quality

11.2.1 Summary of Estimate Quality

Estimates vary in the quality of the assumptions used to create them. Some of these characteristics can be evaluated programmatically. This report rates the quality of the estimate on a 5 point verbal scale:

- Excellent
- Very Good
- Good
- Fair
- Poor

Overall quality of this estimate: Good

11.2.2 Calibration Evaluation

Estimates calibrated with three or more historical projects are most accurate. Estimates calibrated with one or two historical projects, cost drivers, or project types are less accurate. This estimate has been calibrated using project type.

Calibration Quality: Good

11.2.3 Scope Evaluation

Scope estimates created with fine-granularity units such as lines of code include less imprecision than estimates created with large-granularity units such as classes and subsystems.

This estimate's type of scope estimate: Basic Size (Classes/Modules)

Scope Estimate Quality: Good

11.2.4 Phase Evaluation

Estimates created later in a project are accurate than estimates created early in the project. Even the best estimates cannot be very accurate if they are created at a point in the project when comparatively little is known about the software to be built.

Current Project Phase: General Requirements Complete

Estimate quality possible in this phase: Fair

11.2.5 Consistency Check

The table below provides a consistency check by comparing the current project estimate to results from other projects of similar sizes and types. The estimated project has a type of "Business Systems".

Management Metric	Value	Assessment
Productivity (lines of code per staff-month)	2,653	Within Normal Range
Schedule (Months)	11.2	Within Normal Range
Effort (Staff-Months)	15	Within Normal Range
Average Staff (people)	1.3	Within Normal Range
Code Generation Rate (lines of code per month)	3,506	Within Normal Range

Table 11-3: Package 2 Consistency Check

11.2.6 Suitability Evaluation

The Estimate tool works effectively when at least two of the following conditions are met:

- Estimated size is greater than or equal to 5000 lines of code
- Nominal development is expected to be at least 6 months
- Nominal effort is expected to be at least 18 staff-months
- Nominal peak staffing is at least 3 people

When less than two of these criteria are met, the only way to achieve a reliable estimate is to use historical calibration. Even when historical calibration is used, some projects are too small to estimate reliably. This project uses Project type (from industry data) calibration and four of these conditions have been met.

Suitability of Construx Estimate to estimate this project: Excellent

11.3 Planning Options Overview

11.3.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	25
Schedule (calendar months)	9.8
Cost	$277,914
Peak Staff (people)	3.5
Average Staff (people)	2.6

Table 11-4: Package 2 Planning - Nominal

11.3.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	15
Schedule (calendar months)	11.2
Cost	$164,446
Peak Staff (people)	1.8
Average Staff (people)	1.3

Table 11-5: Package 2 Planning - Optimum

11.3.3 Shortest Schedule Plan

Management Metric	Planning Value
Effort (staff months)	61
Schedule (calendar months)	7.8
Cost	$678,500
Peak Staff (people)	10.5
Average Staff (people)	7.8

Table 11-6: Package 2 Planning – Shortest Schedule

11.3.4 Least Cost Plan

Management Metric	Planning Value
Effort (staff months)	9
Schedule (calendar months)	12.7
Cost	$97,305
Peak Staff (people)	0.9
Average Staff (people)	0.7

Table 11-7: Package 2 Planning – Least Cost

11.4 Priorities

Priorities can be used to determine the optimal project plan. The table below shows the priorities used to create this estimate.

Priority	Value
Schedule Priority	High Priority
Effort Priority	High Priority
Cost Priority	High Priority
Peak Staff Priority	Medium Priority

Table 11-8: Package 2 Priorities

11.5 Scope Probabilities

The table below contains scope estimates by probability. These scope estimates are expressed in lines of code. If the scope estimates were not originally expressed by the estimator in lines of code, they have been converted to lines of code. The scope estimates are based on parameters that have been entered by the estimator, including the following:

Scoping Method: Basic Size (Classes/Modules)

Project Phase: General Requirements Complete

Number of Simulations: 500

Probability (%)	Scope Will Be Less Than	Difference From Nominal
1.0	36,448	-7%
5.0	37,113	-5%
10.0	37,556	-4%
20.0	38,110	-3%
30.0	38,443	-2%
40.0	38,665	-1%
50.0	39,108	0%
60.0	39,330	1%
70.0	39,551	1%
80.0	40,106	3%
90.0	40,338	3%
95.0	40,881	5%
99.0	41,879	7%

Table 11-9: Package 2 Scope Probabilities

11.6 Effort Probabilities

The table below contains effort estimates by probability. The effort estimates are expressed in staff-months. They are based on parameters that have been entered by the estimator, including the following:

Calibration Method: Project Type

Project Phase: General Requirements Complete

Number of Simulations: 500

Probability (%)	Effort Will Be Less Than	Difference From Nominal
1.0	10	-61%
5.0	13	-50%
10.0	14	-44%
20.0	17	-33%
30.0	20	-22%
40.0	22	-14%
50.0	25	0%
60.0	30	19%
70.0	34	36%
80.0	43	73%
90.0	62	146%
95.0	112	344%
99.0	506	1,914%

Table 11-10: Package 2 Effort Probabilities

12. Appendix C: ABDAR Software Estimate – Package 3

12.1 Estimate Summary

This document provides the estimate of the remainder of what is considered the minimal functional capability as described in the ABDAR documentation.

12.1.1 Nominal Plan

Current Project Phase: General Requirements Complete

Management Metric	Expected Value (50% Probability)	Standard Deviation	Standard Deviation as Percentage
Lines of Code	338,992	10,141	+/- 3%
Man-Months	459	296	+/- 65%
Schedule (calendar months)	25.8	4.9	+/- 19%
Cost	$5,075,329	$3,275,208	+/- 65%
Peak Staff (People)	27.7	12.3	+/- 45%
Average Staff (People)	17.8	11.5	+/- 65%
Overall Estimate Quality	Good		

Table 12-1: Package 3 Nominal Estimate

This estimate is the 50/50 estimate – the estimate for which there is both a 50 percent chance of overrunning and a 50 percent chance of under running the estimate. This is also known as the nominal estimate. This estimate is for the "main build" phase of a project, the time from detailed requirements specification complete to software acceptance. Earlier phases of a project are not estimated here.

12.1.2 Optimum Plan

Management Metric	Optimum Planning Value
Effort (Staff Months)	272
Schedule (calendar months)	29.4
Cost	$3,003,153
Peak Staff (People)	14.4
Average Staff (People)	9.2

Table 12-2: Package 3 Optimum Estimate

These planning values meet the projects entire set of constraints and priorities to the maximum extent possible.

12.2 Estimate Quality

12.2.1 Summary of Estimate Quality

Estimates vary in the quality of the assumptions used to create them. Some of these characteristics can be evaluated programmatically. This report rates the quality of the estimate on a 5 point verbal scale:

- Excellent
- Very Good
- Good
- Fair
- Poor

Overall quality of this estimate: Good

12.2.2 Calibration Evaluation

Estimates calibrated with three or more historical projects are most accurate. Estimates calibrated with one or two historical projects, cost drivers, or project types are less accurate. This estimate has been calibrated using project type.

Calibration Quality: Good

12.2.3 Scope Evaluation

Scope estimates created with fine-granularity units such as lines of code include less imprecision than estimates created with large-granularity units such as classes and subsystems.

This estimate's type of scope estimate: Basic Size (Classes/Modules)

Scope Estimate Quality: Good

12.2.4 Phase Evaluation

Estimates created later in a project are accurate than estimates created early in the project. Even the best estimates cannot be very accurate if they are created at a point in the project when comparatively little is known about the software to be built.

Current Project Phase: General Requirements Complete

Estimate quality possible in this phase: Fair

12.2.5 Consistency Check

The table below provides a consistency check by comparing the current project estimate to results from other projects of similar sizes and types. The estimated project has a type of "Business Systems".

Management Metric	Value	Assessment
Productivity (lines of code per staff-month)	1,248	Within Normal Range
Schedule (Months)	29.4	Within Normal Range
Effort (Staff-Months)	272	Within Normal Range
Average Staff (people)	9.2	Within Normal Range
Code Generation Rate (lines of code per month)	11,523	Within Normal Range

Table 12-3: Package 3 Consistency Check

12.2.6 Suitability Evaluation

The Estimate tool works effectively when at least two of the following conditions are met:

- . Estimated size is greater than or equal to 5000 lines of code
- Nominal development is expected to be at least 6 months
- Nominal effort is expected to be at least 18 staff-months
- Nominal peak staffing is at least 3 people

When less than two of these criteria are met, the only way to achieve a reliable estimate is to use historical calibration. Even when historical calibration is used, some projects are too small to estimate reliably. This project uses Project type (from industry data) calibration and four of these conditions have been met.

Suitability of Construx Estimate to estimate this project: Excellent

12.3 Planning Options Overview

12.3.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	459
Schedule (calendar months)	25.8
Cost	$5,075,329
Peak Staff (people)	27.7
Average Staff (people)	17.8

Table 12-4: Package 3 Planning - Nominal

12.3.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	272
Schedule (calendar months)	29.4
Cost	$3,003,153
Peak Staff (people)	14.4
Average Staff (people)	9.2

Table 12-5: Package 3 Planning - Optimum

12.3.3 Shortest Schedule Plan

Management Metric	Planning Value
Effort (staff months)	1,121
Schedule (calendar months)	20.6
Cost	$12,390,938
Peak Staff (people)	84.4
Average Staff (people)	68.6

Table 12-6: Package 3 Planning – Shortest Schedule

12.3.4 Least Cost Plan

Management Metric	Planning Value
Effort (staff months)	161
Schedule (calendar months)	33.5
Cost	$1,777,013
Peak Staff (people)	7.5
Average Staff (people)	4.8

Table 12-7: Package 3 Planning – Least Cost

12.4 Priorities

Priorities can be used to determine the optimal project plan. The table below shows the priorities used to create this estimate.

Priority	Value
Schedule Priority	High Priority
Effort Priority	High Priority
Cost Priority	High Priority
Peak Staff Priority	High Priority

Table 12-8: Package 3 Priorities

12.5 Scope Probabilities

The table below contains scope estimates by probability. These scope estimates are expressed in lines of code. If the scope estimates were not originally expressed by the estimator in lines of code, they have been converted to lines of code. The scope estimates are based on parameters that have been entered by the estimator, including the following:

Scoping Method: Basic Size (Classes/Modules)

Project Phase: General Requirements Complete

Number of Simulations: 500

Probability (%)	Scope Will Be Less Than	Difference From Nominal
1.0	315,812	-7%
5.0	323,539	-5%
10.0	328,368	-3%
20.0	331,265	-2%
30.0	335,129	-1%
40.0	337,060	-1%
50.0	338,992	0%
60.0	342,855	1%
70.0	345,753	2%
80.0	348,650	3%
90.0	352,514	4%
95.0	354,445	5%
99.0	360,240	6%

Table 12-9: Package 3 Scope Probabilities

12.6 Effort Probabilities

The table below contains effort estimates by probability. The effort estimates are expressed in staff-months. They are based on parameters that have been entered by the estimator, including the following:

Calibration Method: Project Type

Project Phase: General Requirements Complete

Number of Simulations: 500

Probability (%)	Effort Will Be Less Than	Difference From Nominal
1.0	171	-63%
5.0	207	-55%
10.0	250	-45%
20.0	300	-35%
30.0	350	-24%
40.0	403	-12%
50.0	459	0%
60.0	544	18%
70.0	643	40%
80.0	776	69%
90.0	1,079	135%
95.0	1,666	263%
99.0	7,317	1,494%

Table 12-10: Package 3 Effort Probabilities

13. Appendix D: ABDAR Software Estimate – Package 4

13.1 Estimate Summary

This document provides the estimate of the remainder of what is considered the minimal functional capability as described in the ABDAR documentation.

13.1.1 Nominal Plan

Current Project Phase: Feasibility Study/Product Concept Complete

Management Metric	Expected Value (50% Probability)	Standard Deviation	Standard Deviation as Percentage
Lines of Code	182,242	7,671	+/- 4%
Man-Months	154	113	+/- 73%
Schedule (calendar months)	17.1	3.3	+/- 19%
Cost	$1,640,431	$1,205,312	+/- 73%
Peak Staff (People)	14.0	7.3	+/- 52%
Average Staff (People)	9.0	6.6	+/- 73%
Overall Estimate Quality	Good		

Table 13-1: Package 4 Nominal Estimate

This estimate is the 50/50 estimate – the estimate for which there is both a 50 percent chance of overrunning and a 50 percent chance of under running the estimate. This is also known as the nominal estimate. This estimate is for the "main build" phase of a project, the time from detailed requirements specification complete to software acceptance. Earlier phases of a project are not estimated here.

13.1.2 Optimum Plan

Management Metric	Optimum Planning Value
Effort (Staff Months)	91
Schedule (calendar months)	19.5
Cost	$970,669
Peak Staff (People)	7.3
Average Staff (People)	4.7

Table 13-2: Package 4 Optimum Estimate

These planning values meet the projects entire set of constraints and priorities to the maximum extent possible.

13.2 Estimate Quality

13.2.1 Summary of Estimate Quality

Estimates vary in the quality of the assumptions used to create them. Some of these characteristics can be evaluated programmatically. This report rates the quality of the estimate on a 5 point verbal scale:

- Excellent
- Very Good
- Good
- Fair
- Poor

Overall quality of this estimate: Good

13.2.2 Calibration Evaluation

Estimates calibrated with three or more historical projects are most accurate. Estimates calibrated with one or two historical projects, cost drivers, or project types are less accurate. This estimate has been calibrated using project type.

Calibration Quality: Good

13.2.3 Scope Evaluation

Scope estimates created with fine-granularity units such as lines of code include less imprecision than estimates created with large-granularity units such as classes and subsystems.

This estimate's type of scope estimate: Basic Size (Classes/Modules)

Scope Estimate Quality: Good

13.2.4 Phase Evaluation

Estimates created later in a project are accurate than estimates created early in the project. Even the best estimates cannot be very accurate if they are created at a point in the project when comparatively little is known about the software to be built.

Current Project Phase: General Requirements Complete

Estimate quality possible in this phase: Fair

13.2.5 Consistency Check

The table below provides a consistency check by comparing the current project estimate to results from other projects of similar sizes and types. The estimated project has a type of "Business Systems".

Management Metric	Value	Assessment
Productivity (lines of code per staff-month)	1,998	Within Normal Range
Schedule (Months)	19.5	Within Normal Range
Effort (Staff-Months)	91	Within Normal Range
Average Staff (people)	4.7	Within Normal Range
Code Generation Rate (lines of code per month)	9.325	Within Normal Range

Table 13-3: Package 4 Consistency Check

13.2.6 Suitability Evaluation

The Estimate tool works effectively when at least two of the following conditions are met:

- Estimated size is greater than or equal to 5000 lines of code
- Nominal development is expected to be at least 6 months
- Nominal effort is expected to be at least 18 staff-months
- Nominal peak staffing is at least 3 people

When less than two of these criteria are met, the only way to achieve a reliable estimate is to use historical calibration. Even when historical calibration is used, some projects are too small to estimate reliably. This project uses Project type (from industry data) calibration and four of these conditions have been met.

Suitability of Construx Estimate to estimate this project: Excellent

13.3 Planning Options Overview

13.3.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	154
Schedule (calendar months)	17.1
Cost	$1,640,431
Peak Staff (people)	14.0
Average Staff (people)	9.0

Table 13-4: Package 4 Planning - Nominal

13.3.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	91
Schedule (calendar months)	19.5
Cost	$970,669
Peak Staff (people)	7.3
Average Staff (people)	4.7

Table 13-5: Package 4 Planning - Optimum

13.3.3 Shortest Schedule Plan

Management Metric	Planning Value
Effort (staff months)	376
Schedule (calendar months)	13.7
Cost	$4,004,958
Peak Staff (people)	42.7
Average Staff (people)	27.4

Table 13-6: Package 4 Planning – Shortest Schedule

13.3.4 Least Cost Plan

Management Metric	Planning Value
Effort (staff months)	54
Schedule (calendar months)	22.3
Cost	$574,360
Peak Staff (people)	3.8
Average Staff (people)	2.4

Table 13-7: Package 4 Planning – Least Cost

13.4 Priorities

Priorities can be used to determine the optimal project plan. The table below shows the priorities used to create this estimate.

Priority	Value
Schedule Priority	High Priority
Effort Priority	High Priority
Cost Priority	High Priority
Peak Staff Priority	Med Priority

Table 13-8: Package 4 Priorities

13.5 Scope Probabilities

The table below contains scope estimates by probability. These scope estimates are expressed in lines of code. If the scope estimates were not originally expressed by the estimator in lines of code, they have been converted to lines of code. The scope estimates are based on parameters that have been entered by the estimator, including the following:

Scoping Method: Basic Size (Classes/Modules)

Project Phase: General Requirements Complete

Number of Simulations: 500

Probability (%)	Scope Will Be Less Than	Difference From Nominal
1.0	166,092	-9%
5.0	170,937	-6%
10.0	173,359	-5%
20.0	175,782	-4%
30.0	179,012	-2%
40.0	180,627	-1%
50.0	182,242	0%
60.0	185,472	2%
70.0	187,894	3%
80.0	189,509	4%
90.0	193,547	6%
95.0	195,162	7%
99.0	198,392	9%

Table 13-9: Package 4 Scope Probabilities

13.6 Effort Probabilities

The table below contains effort estimates by probability. The effort estimates are expressed in staff-months. They are based on parameters that have been entered by the estimator, including the following:

Calibration Method: Project Type

Project Phase: General Requirements Complete

Number of Simulations: 500

Probability (%)	Effort Will Be Less Than	Difference From Nominal
1.0	51	-67%
5.0	68	-56%
10.0	81	-47%
20.0	101	-34%
30.0	119	-23%
40.0	136	-12%
50.0	154	0%
60.0	180	17%
70.0	215	40%
80.0	266	72%
90.0	466	202%
95.0	658	327%
99.0	2,138	1,287%

Table 13-10: Package 4 Effort Probabilities

14. Appendix E: ABDAR Software Estimate – Package 5

14.1 Estimate Summary

This summary covers the effort required for the following activities:

- Creation and modification of a database (by aircraft) from TO source data
- Modification to the original ABDAR system to extract data from the database as opposed to either manual input or copy/paste from the original TO

14.1.1 Nominal Plan

Current Project Phase: Feasibility Study/Project Concept Complete

Management Metric	Expected Value (50% Probability)	Standard Deviation	Standard Deviation as Percentage
Lines of Code	65,708	2,138	+/- 3%
Man-Months	57	39	+/- 69%
Schedule (calendar months)	12.9	2.5	+/- 19%
Cost	$631,456	$435,207	+/- 69%
Peak Staff (People)	6.3	3.0	+/- 47%
Average Staff (People)	4.4	3.1	+/- 69%
Overall Estimate Quality	Good		

Table 14-1: Package 5 Nominal Estimate

This estimate is the 50/50 estimate – the estimate for which there is both a 50 percent chance of overrunning and a 50 percent chance of under running the estimate. This is also known as the nominal estimate. This estimate is for the "main build" phase of a project, the time from detailed requirements specification complete to software acceptance. Earlier phases of a project are not estimated here.

14.1.2 Optimum Plan

Management Metric	Optimum Planning Value
Effort (Staff Months)	34
Schedule (calendar months)	14.7
Cost	$373,642
Peak Staff (People)	3.3
Average Staff (People)	2.3

Table 14-2: Package 5 Optimum Estimate

These planning values meet the projects entire set of constraints and priorities to the maximum extent possible.

14.2 Estimate Quality

14.2.1 Summary of Estimate Quality

Estimates vary in the quality of the assumptions used to create them. Some of these characteristics can be evaluated programmatically. This report rates the quality of the estimate on a 5 point verbal scale:

- Excellent
- Very Good
- Good
- Fair
- Poor

Overall quality of this estimate: Good

14.2.2 Calibration Evaluation

Estimates calibrated with three or more historical projects are most accurate. Estimates calibrated with one or two historical projects, cost drivers, or project types are less accurate. This estimate has been calibrated using project type.

Calibration Quality: Good

14.2.3 Scope Evaluation

Scope estimates created with fine-granularity units such as lines of code include less imprecision than estimates created with large-granularity units such as classes and subsystems.

This estimate's type of scope estimate: Basic Size (Classes/Modules)

Scope Estimate Quality: Good

14.2.4 Phase Evaluation

Estimates created later in a project are accurate than estimates created early in the project. Even the best estimates cannot be very accurate if they are created at a point in the project when comparatively little is known about the software to be built.

Current Project Phase: Feasibility Study/Product Concept Complete

Estimate quality possible in this phase: Fair

14.2.5 Consistency Check

The table below provides a consistency check by comparing the current project estimate to results from other projects of similar sizes and types. The estimated project has a type of "Intranet Systems (internal)".

Management Metric	Value	Assessment
Productivity (lines of code per staff-month)	1,945	Within Normal Range
Schedule (Months)	14.7	Within Normal Range
Effort (Staff-Months)	34	Within Normal Range
Average Staff (people)	2.3	Within Normal Range
Code Generation Rate (lines of code per month)	4,472	Within Normal Range

Table 14-3: Package 5 Consistency Check

14.2.6 Suitability Evaluation

The Estimate tool works effectively when at least two of the following conditions are met:

- Estimated size is greater than or equal to 5000 lines of code
- Nominal development is expected to be at least 6 months
- Nominal effort is expected to be at least 18 staff-months
- Nominal peak staffing is at least 3 people

When less than two of these criteria are met, the only way to achieve a reliable estimate is to use historical calibration. Even when historical calibration is used, some projects are too small to estimate reliably. This project uses Project type (from industry data) calibration and four of these conditions have been met.

Suitability of Construx Estimate to estimate this project: Excellent

14.3 Planning Options Overview

14.3.1 Nominal Plan

Management Metric	Planning Value
Effort (staff months)	57
Schedule (calendar months)	12.9
Cost	$631,456
Peak Staff (people)	6.3
Average Staff (people)	4.4

Table 14-4: Package 5 Planning - Nominal

14.3.2 Optimum Plan

Management Metric	Planning Value
Effort (staff months)	34
Schedule (calendar months)	14.7
Cost	$373,642
Peak Staff (people)	3.3
Average Staff (people)	2.3

Table 14-5: Package 5 Planning - Optimum

14.3.3 Shortest-Schedule Plan

Management Metric	Planning Value
Effort (staff months)	139
Schedule (calendar months)	10.3
Cost	$1,541,641
Peak Staff (people)	19.3
Average Staff (people)	13.5

Table 14-6: Package 5 Planning – Shortest Schedule

14.3.4 Least-Cost Plan

Management Metric	Planning Value
Effort (staff months)	20
Schedule (calendar months)	16.8
Cost	$221,090
Peak Staff (people)	1.7
Average Staff (people)	1.2

Table 14-7: Package 5 Planning – Least Cost

14.4 Priorities

Priorities can be used to determine the optimal project plan. The table below shows the priorities used to create this estimate.

Priority	Value
Schedule Priority	High Priority
Effort Priority	High Priority
Cost Priority	High Priority
Peak Staff Priority	High Priority

Table 14-8: Package 5 Priorities

14.5 Scope Probabilities

The table below contains scope estimates by probability. These scope estimates are expressed in lines of code. If the scope estimates were not originally expressed by the estimator in lines of code, they have been converted to lines of code. The scope estimates are based on parameters that have been entered by the estimator, including the following:

Scoping Method: Basic Size (Classes/Modules)

Project Phase: Feasibility Study/Product Concept Complete

Number of Simulations: 500

Probability (%)	Scope Will Be Less Than	Difference From Nominal
1.0	60,958	-7%
5.0	61,908	-6%
10.0	63,095	-4%
20.0	63,808	-3%
30.0	64,758	-1%
40.0	65,233	-1%
50.0	65,708	0%
60.0	66,183	1%
70.0	67,370	3%
80.0	67,845	3%
90.0	68,558	4%
95.0	69,983	7%
99.0	70,933	8%

Table 14-9: Package 5 Scope Probabilities

14.6 Effort Probabilities

The table below contains effort estimates by probability. The effort estimates are expressed in staff-months. They are based on parameters that have been entered by the estimator, including the following:

Calibration Method: Project Type

Project Phase: Feasibility Study/Product Concept Complete

Number of Simulations: 500

Probability (%)	Effort Will Be Less Than	Difference From Nominal
1.0	21	-64%
5.0	28	-50%
10.0	33	-42%
20.0	41	-28%
30.0	46	-20%
40.0	50	-12%
50.0	57	0%
60.0	66	15%
70.0	81	42%
80.0	103	80%
90.0	155	171%
95.0	313	448%
99.0	2,766	4,744%

Table 14-10: Package 5 Effort Probabilities

15. Appendix F: Software Evaluation Criteria

15.1 Software Quality Criteria

Name	Definition	Method				Scale		Source
		D	T	A	I	AFRL	MSG	
1. Functional Analysis								[SEI]
1. Effectiveness	The degree to which a system's features and capabilities meet the user's needs.	X	X	X	X			[SEI]
1. Necessity of Characteristics	The degree to which all of the necessary features and capabilities are present in the software system.	X			X			[SEI]
2. Sufficiency of Characteristics	The degree to which the features and capabilities of a software system adequately meet the user's needs.	X	X	X	X			[SEI]
2. Responsiveness	The degree to which a software system or component has incorporated the user's requirements.	X	X	X	X			[SEI]
3. Correctness	The degree to which a system or component is free from faults in its specification, design, and implementation.	X	X	X	X			[IEEE]
1. Completeness/Incompleteness	The degree to which all the parts of a software system or component are present and each of its parts is fully specified and developed.				X			[Boehm]
2. Consistency	The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a system or component.				X			[IEEE]
3. Traceability	The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another.				X			[IEEE]
4. Provably Correct	The ability to mathematically verify the correctness of a system or component.		X	X				[SEI]
4. Verifiability	The relative effort to verify the specified software operation and performance.			X				[Evans]
1. Testability	The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met [IEEE]. Note: Not only is testability a measurement for software, it can also apply to the testing scheme.	X			X			[IEEE]

Name	Definition	Method				Scale		Source
		D	T	A	I	AFRL	MSG	
2. Human Factors								[SEI]
1. Usability	The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.	X						[IEEE]
1. Error Proneness	The degree to which a system may allow the user to intentionally or unintentionally introduce errors into or misuse the system.	X						[SEI]
2. Operability	The ease of operating the software.	X						[Deutsch]
2. Fidelity	The degree of similarity between a model and the system properties being modeled.	X			X			[IEEE]
3.Technical Design								[SEI]
1. Dependability	The property of a computer system such that reliance can justifiably be placed on the service it delivers.	X	X					[Barbacci]
1. Availability/Robustness	The degree to which a system or component is operational and accessible when required for use.	X	X					[IEEE]
- Error Tolerance	The ability of a system or component to continue normal operation despite the presence of erroneous inputs.	X	X					[IEEE]
- Fault Tolerance	The ability of a system or component to continue normal operation despite the presence of hardware or software faults.	X	X					[IEEE]
- Fail Safe	Pertaining to a system or component that automatically places itself in a safe operating mode in the event of a failure.	X	X					[IEEE]
- Fail Soft	Pertaining to a system or component that continues to provide partial operational capability in the event of certain failures.	X	X					[IEEE]
2. Reliability	The ability of a system or component to perform its required functions under stated conditions for a specified period of time.	X	X					[IEEE]
1. Accuracy	A quantitative measure of the magnitude of error.	X	X	X				[IEEE]
3. Trustworthiness	The degree to which a system or component avoids compromising, corrupting, or delaying sensitive information.	X	X	X				[SEI]
1. Vulnerability	The degree to which a software system or component is open to unauthorized access, change, or disclosure of information and is susceptible to interference or disruption of system services.	X	X	X				[SEI]

Name	Definition	Method				Scale		Source
		D	T	A	I	AFRL	MSG	
1. Integrity	The degree to which a system or component prevents unauthorized access to, or modification of, computer programs or data.	X	X					[IEEE]
2. Confidentiality	The nonoccurrence of the unauthorized disclosure of information.	X	X	X				[Barbacci]
- Anonymity	The degree to which a software system or component allows for or supports anonymous transactions.	X	X					[SEI]
3. Denial of Service	The degree to which a software system or component prevents the interference or disruption of system services to the user.	X	X					[SEI]
- Accessibility	The degree to which the software system protects system functions or service from being denied to the user.	X	X					[SEI]
4. Survivability	The degree to which essential functions are still available even though some part of the system is down.	X	X					[Deutsch]
2. Accountability	The degree to which the software system allows the tracking and monitoring of updates or changes.	X	X					
1. Auditable	The degree to which a software system records information concerning transactions performed against the system.	X	X	X	X			[SEI]
4. Security	The ability of a system to manage, protect, and distribute sensitive information.	X	X					[SEI]
2. Efficiency/Resource Utilization	The degree to which a system or component performs its designated functions with minimum consumption of resources (CPU, Memory, I/O, Peripherals, Networks).	X	X	X	X			[IEEE]
- Speed	The rate at which a software system or component performs its functions.	X	X	X				[SEI]
- Compactness	The degree to which a system or component makes efficient use of its data storage space- occupies a small volume.	X	X	X				[SEI]
1. Capacity	A measure of the amount of work a system can perform.	X	X	X				[Barbacci]
2. Real-time Responsiveness/Latency	The ability of a system or component to respond to an inquiry or demand within a prescribed time frame.	X	X					[SEI]
3. Throughput	The amount of work that can be performed by a computer system or component in a given period of time.	X	X					[IEEE]
3. Maintainability	The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.	X	X		X			[IEEE]
- Modifiability	The degree to which a system or component facilitates the	X	X		X			[Boehm]

Name	Definition	Method				Scale		Source
		D	T	A	I	AFRL	MSG	
	incorporation of changes, once the nature of the desired change has been determined.							
- Flexibility/Adaptability	The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed.	X	X		X			[IEEE]
- Evolvability/Upgradeability	The ease with which a system or component can be modified to take advantage of new software or hardware technologies.	X	X		X			[SEI]
- Extendibility/Expandability	The ease with which a system or component can be modified to increase its storage or functional capacity.	X	X		X			[IEEE]
4. Understandability	The degree to which the purpose of the system or component is clear to the evaluator.	X	X		X			[Boehm]
1. Complexity	The overall measurement of which the system or component is understandable or verifiable.	X	X					
- Apparent	The degree to which a system or component has a design or implementation that is difficult to understand and verify.	X			X			[IEEE]
- Inherent	The degree of complication of a system or system component, determined by such factors as the number and intricacy of interfaces, the number and intricacy of conditional branches, the degree of nesting, and the types of data structures.	X	X					[Evans]
2. Simplicity	The degree to which a system or component has a design and implementation that is straightforward and easy to understand.	X			X			[IEEE]
3. Structuredness	The degree to which a system or component possesses a definite pattern of organization of its interdependent parts.	X			X			[Boehm]
4. Readability	The degree to which a system's functions and those of its component statements can be easily discerned by reading the associated source code.	X			X			[SEI]
- Self-Descriptiveness	The degree to which a system or component contains enough information to explain its objectives and properties.	X			X			[IEEE]
- Conciseness	The degree to which a software system or component has no excessive information present.	X			X			
4. Implementation								
1. Interoperability	The ability of two or more systems or components to exchange information and to use the information that has been exchanged.	X	X					[IEEE]

Name	Definition	Method				Scale		Source
		D	T	A	I	AFRL	MSG	
1. Compatibility	The ability of two or more systems or components to perform their required functions while sharing the same hardware or software environment.	X	X					[IEEE]
2. Openness	The degree to which a system or component complies with standards.	X			X			[SEI]
- Commonality	The degree to which standards are used to achieve interoperability.	X			X			[SEI]
2. Portability	The ease with which a system or component can be transferred from one hardware or software environment to another.	X	X					[IEEE]
3. Scalability	The ease with which a system or component can be modified to fit the problem area.	X	X		X			
4. Reusability	The degree to which a software module or other work product can be used in more than one computing program or software system.	X	X		X			[IEEE]
1. Functional Scope	The range or scope to which a system component is capable of being applied.	X	X		X			[SEI]
- Generality	The measurement to which this component is capable of being applied.	X	X		X			
- Abstractness	The degree to which a system or component performs only the necessary functions relevant to a particular purpose.	X	X		X			[SEI]
- Accessibility	The degree to which a software system or component facilitates the selective use of its components.	X	X		X			[Boehm]
2. Retrievability								
5. Cost of Ownership	The overall cost of a computer system to an organization to include the costs associated with operating and maintaining the system, and the lifetime of operational use of the system.	X		X				[SEI]
1. Cost of Operation	The overall cost of operating a computer system to include the costs associated with personnel, training, and system operations.	X		X				[SEI]
1. Operations Personnel	The number of personnel needed to operate all aspects of a computer system, including the support personnel and facilities needed to support that activity.	X		X				[SEI]
2. Training	Provisions to learn how to develop, maintain, or use the software system.	X		X				[SEI]
3. Operations system	The cost of environmentals, communication, licenses,	X		X				[SEI]

Name	Definition	Method				Scale		Source
		D	T	A	I	AFRL	MSG	
	expendables, and documentation maintenance for an operational system.							
2. Cost of maintenance	The overall cost of maintaining a computer system to include the costs associated with personnel, training, maintenance control, hardware and software maintenance, and requirements growth.			X				[SEI]
1. Maintenance Personnel	The number of personnel needed to maintain all aspects of a computer system, including the support personnel and facilities needed to support that activity.			X				[SEI]
2. Training	Provisions to learn how to develop, maintain, or use the software system.			X				[SEI]
3. Maintenance Control	The cost of planning and scheduling hardware preventive maintenance, and software maintenance and upgrades, managing the hardware and software baselines, and providing response for hardware corrective maintenance.			X				[SEI]
4. Hardware Maintenance	The cost of maintenance of the hardware.			X				[SEI]
5. Software Maintenance	The cost associated with modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment.			X				[SEI]
6. Requirements Growth	The rate at which the requirements change for an operational system. The rate can be positive or negative.			X				[SEI]
3. Lifetime of Operational Capability	The measurement of the time required for the acquisition and/or change of the system.			X				[SEI]
1. Acquisition Cycle Time	The period of time that starts when a system is conceived and ends when the product meets its initial operational capability.			X				[SEI]
2. Software Change Cycle Time	The period of time that starts when a new system requirement is identified and ends when the requirement has been incorporated into the system and delivered for operational use.			X				[SEI]
6. Productivity	The quality or state of being productive.			X				[Webster]

15.2 Software Quality Method

Type	Definition
Demonstration (D)	Demonstration is verification of the operation of the system that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis. Demonstration of avionics and hardware involves operation, movement and/or adjustment of the item in performing its designed functions, under a specific set of conditions, without recording quantitative data. Demonstration of software is defined as verification that a specification requirement has been met, where success is determined by direct observation.
Test (T)	Test is verification of the system using instrumentation or other special test equipment. Test of software is defined as verification that a specification requirement is met by a thorough exercising of the applicable element under appropriate conditions for the collection of data that may be analyzed to prove compliance. Functional test requirements may be satisfied by acceptance test plans, procedures, test results/reports.
Analysis (A)	Analysis is verification of the system by processing accumulated data obtained from other qualification methods. Analysis of software is defined as verification that a specification requirement has been met by technical evaluation of equations or design. Verification of software may also involve processing or analyzing data resulting from software test results.
Inspection (I)	Inspection is a visual examination of system components, documentation, etc.

16. Appendix G: ABDAR Software Implementation Criteria

Final Grades based on criteria:

AFRL: 21.2 MSG: 11.4

Criteria Matrix Legend:

High	These features are mandatory. The application must include this feature.
(Scale: >50-100)	
Moderate	Features would be nice to have, or they may even be necessary but with loose guidelines on how they are implemented.
(Scale: >11-50)	
Low	These features are not necessary.
(Scale: 1-11)	

ABDAR Software Implementation Criteria

				Weight					AFRL					MSG				
				0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000
				Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present
	Weight	Description		e	d	c	b	a	e	d	c	b	a	e	d	c	b	a
Functional Analysis – Effectiveness	100.0	A	a	10.0	32.5	55.0	77.5	100.0	■		■	■	■	■		■	■	■
	77.5	B	b	7.8	25.2	42.6	60.1	77.5	■			■	■	■			■	■
	55.0	C	c	5.5	17.9	30.3	42.6	55.0	■				■	■				■
	32.5	D	d	3.3	10.6	17.9	25.2	32.5	■	■			32.5	■	■			
	10.0	F	e	1.0	3.3	5.5	7.8	10.0	■	■	■	■	■	■	■	■	■	10.0
Functional Analysis – Responsiveness	100.0	A	a	10.0	32.5	55.0	77.5	100.0	■		■	■	■	■		■	■	■
	77.5	B	b	7.8	25.2	42.6	60.1	77.5	■			■	■	■			■	■
	55.0	C	c	5.5	17.9	30.3	42.6	55.0	■				■	■				■
	32.5	D	d	3.3	10.6	17.9	25.2	32.5	■	■				■	■			
	10.0	F	e	1.0	3.3	5.5	7.8	10.0	■	■	■	■	10.0	■	■	■	■	10.0

ABDAR Software Implementation Criteria

					Weight					AFRL					MSG				
				0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000
				Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present
	Weight	Description		e	d	c	b	a	e	d	c	b	a	e	d	c	b	a
Functional Analysis (cont.) Correctness	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
	77.5	B	b	7.8	25.2	42.6	60.1	77.5										
	55.0	C	c	5.5	17.9	30.3	42.6	55.0										
	32.5	D	d	3.3	10.6	17.9	25.2	32.5					32.5					
	10.0	F	e	1.0	3.3	5.5	7.8	10.0										10.0
Verifiability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
	77.5	B	b	7.8	25.2	42.6	60.1	77.5										
	55.0	C	c	5.5	17.9	30.3	42.6	55.0										
	32.5	D	d	3.3	10.6	17.9	25.2	32.5										
	10.0	F	e	1.0	3.3	5.5	7.8	10.0					10.0					10.0

ABDAR Software Implementation Criteria					Weight					AFRL					MSG				
					0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000
					Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present
		Weight	Description		e	d	c	b	a	e	d	c	b	a	e	d	c	b	a
Human Factors	Usability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
		77.5	B	b	7.8	25.2	42.6	60.1	77.5										
		55.0	C	c	5.5	17.9	30.3	42.6	55.0										
		32.5	D	d	3.3	10.6	17.9	25.2	32.5									25.2	
		10.0	F	e	1.0	3.3	5.5	7.8	10.0				7.8						
	Fidelity	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
		77.5	B	b	7.8	25.2	42.6	60.1	77.5										
		55.0	C	c	5.5	17.9	30.3	42.6	55.0										
		32.5	D	d	3.3	10.6	17.9	25.2	32.5					32.5					
		10.0	F	e	1.0	3.3	5.5	7.8	10.0										10.0

ABDAR Software Implementation Criteria

					Weight					AFRL					MSG			
				0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000
				Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present
	Weight	Description		e	d	c	b	a	e	d	c	b	a	e	d	c	b	a
Technical Design Dependability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
	77.5	B	b	7.8	25.2	42.6	60.1	77.5										
	55.0	C	c	5.5	17.9	30.3	42.6	55.0										
	32.5	D	d	3.3	10.6	17.9	25.2	32.5										
	10.0	F	e	1.0	3.3	5.5	7.8	10.0					10.0					10.0
Technical Design Efficiency/Resource Utilization	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
	77.5	B	b	7.8	25.2	42.6	60.1	77.5										
	55.0	C	c	5.5	17.9	30.3	42.6	55.0			30.3							
	32.5	D	d	3.3	10.6	17.9	25.2	32.5										
	10.0	F	e	1.0	3.3	5.5	7.8	10.0								5.5		

ABDAR Software Implementation Criteria					Weight					AFRL					MSG				
					0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000
					Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present
		Weight	Description		e	d	c	b	a	e	d	c	b	a	e	d	c	b	a
Technical Design (cont.)	Maintainability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
		77.5	B	b	7.8	25.2	42.6	60.1	77.5										
		55.0	C	c	5.5	17.9	30.3	42.6	55.0										
		32.5	D	d	3.3	10.6	17.9	25.2	32.5										
		10.0	F	e	1.0	3.3	5.5	7.8	10.0					10.0					10.0
	Understandability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
		77.5	B	b	7.8	25.2	42.6	60.1	77.5										
		55.0	C	c	5.5	17.9	30.3	42.6	55.0										
		32.5	D	d	3.3	10.6	17.9	25.2	32.5										
		10.0	F	e	1.0	3.3	5.5	7.8	10.0					10.0					10.0

ABDAR Software Implementation Criteria					Weight					AFRL					MSG				
					0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000
					Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present
		Weight	Description		e	d	c	b	a	e	d	c	b	a	e	d	c	b	a
Implementation	Interoperability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
		77.5	B	b	7.8	25.2	42.6	60.1	77.5										
		55.0	C	c	5.5	17.9	30.3	42.6	55.0										
		32.5	D	d	3.3	10.6	17.9	25.2	32.5			17.9							
		10.0	F	e	1.0	3.3	5.5	7.8	10.0								5.5		
	Portability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
		77.5	B	b	7.8	25.2	42.6	60.1	77.5										
		55.0	C	c	5.5	17.9	30.3	42.6	55.0										
		32.5	D	d	3.3	10.6	17.9	25.2	32.5	3.3									
		10.0	F	e	1.0	3.3	5.5	7.8	10.0						1.0				

ABDAR Software Implementation Criteria

					Weight					AFRL					MSG			
				0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000
				Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present
	Weight	Description		e	d	c	b	a	e	d	c	b	a	e	d	c	b	a
Implementation (cont.) Scalability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
	77.5	B	b	7.8	25.2	42.6	60.1	77.5										
	55.0	C	c	5.5	17.9	30.3	42.6	55.0										
	32.5	D	d	3.3	10.6	17.9	25.2	32.5										
	10.0	F	e	1.0	3.3	5.5	7.8	10.0			5.5					5.5		
Reusability	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
	77.5	B	b	7.8	25.2	42.6	60.1	77.5										
	55.0	C	c	5.5	17.9	30.3	42.6	55.0										
	32.5	D	d	3.3	10.6	17.9	25.2	32.5			17.9							
	10.0	F	e	1.0	3.3	5.5	7.8	10.0								5.5		

ABDAR Software Implementation Criteria

					Weight					AFRL					MSG				
				0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000	0.100	0.325	0.550	0.775	1.000
				Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present	Do not need to have	Nice to have	Somewhat necessary	Very desirable	Has to be present
	Weight	Description		e	d	c	b	a	e	d	c	b	a	e	d	c	b	a
Implementation (cont.) Cost of Ownership	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
	77.5	B	b	7.8	25.2	42.6	60.1	77.5										
	55.0	C	c	5.5	17.9	30.3	42.6	55.0										
	32.5	D	d	3.3	10.6	17.9	25.2	32.5										
	10.0	F	e	1.0	3.3	5.5	7.8	10.0			5.5					5.5		
Implementation (cont.) Productivity	100.0	A	a	10.0	32.5	55.0	77.5	100.0										
	77.5	B	b	7.8	25.2	42.6	60.1	77.5										
	55.0	C	c	5.5	17.9	30.3	42.6	55.0										
	32.5	D	d	3.3	10.6	17.9	25.2	32.5					32.5					
	10.0	F	e	1.0	3.3	5.5	7.8	10.0										10.0

17. Appendix H: Requirements Traceability Matrix

Package	Description	Requirements
17.1 Package 1	Provide an electronic AFTO 97 Form	3.2.4.1.1-5 The ABDAR system shall provide the ability to precisely record damage site locations and assign damage site identification numbers.
		3.2.4.1.2-36 The ABDAR system shall provide the capability to input the damage dimensions.
		3.2.4.1.3-8 The ABDAR system shall provide a method to designate which repairs require re-inspection.
		3.2.4.3-1 The ABDAR system shall present tasks and record repair accomplishments in order to execute the directed repair plan.
		3.2.4.3-6 The ABDAR system shall provide technical data and process controls needed to inspect repairs and document inspections.
		3.2.4.3-7 The ABDAR system shall provide a means to record results of inspections and functional tests.
		3.2.5.1-1 The ABDAR system shall collect and record debrief materials.
		3.2.5.1-2 The ABDAR system shall provide for electronic transfer or print-out of debrief information.
		3.2.5.1-3 The ABDAR system shall provide for signing off 'UXO clear'.
		3.2.5.1-4 The ABDAR system shall provide for collection and storage of assessment documentation information as currently recorded on Forms 97, 97A and 781.
		3.2.5.1-5 The ABDAR system shall provide the capability to print assessment documentation information.
		3.2.5.2-3 The ABDAR system shall provide the capability to print documentation of all ABDAR maintenance.

Package	Description	Requirements
		3.2.5.2-5 The ABDAR system shall provide for collecting and storing documentation of all ABDR maintenance.
		3.2.6.2-5 The ABDAR system shall operate in stand-alone mode to deliver assessment and repair information.
		3.2.6.2-14 The ABDAR system shall provide the capability to mark the point where the ABDAR process was exited.
	Provide wizards to guide user through process of performing ABDR assessment.	3.2.2.2-5 The ABDAR system shall implement a capability to aid in performing ABDR related portions of the aircrew debrief, including a question set for aircrew debrief.
		3.2.4.1.1-1 The ABDAR system shall provide procedures for ensuring an A/C is Safe for Maintenance.
		3.2.4.1.1-2 The ABDAR system shall provide procedures for inspecting for UXO.
		3.2.4.1.2-3 The ABDAR system shall provide procedures for performing damage assessment.
	Digital TO access provided by third party viewers such as plug-ins in a web browser.	3.2.4.1.2-6 The ABDAR system shall present technical data needed to perform functional tests.
		3.2.4.1.2-7 The ABDAR system shall present technical data needed to inspect damage sites.
		3.2.4.1.2-13 The ABDAR system shall provide the assessor the means to identify the subsystems installed in any location in the A/C.
		3.2.4.1.2-27 The ABDAR system shall provide access to flight restriction information.
		3.2.4.1.2.-30 The ABDAR system shall provide direct access to specific elements of information on A/C systems or components.

Package	Description	Requirements
		3.2.4.1.2-31 The ABDAR system shall provide direct access to information on wiring.
		3.2.4.1.2-33 The ABDAR system shall identify the minimum requirements for redundant systems and/or redundant functions within systems.
		3.2.4.1.2-34 The ABDAR system shall provide access to information in TO 1-1A-8, Aircraft and Missile Repair, Structural Hardware, and NAVAIR 01-1A-20, Structural Hardware.
		3.2.4.1.3-10 The ABDAR system shall provide access to information on A/C structures.
		3.2.4.1.3-11 The ABDAR system shall provide access to information on composite materials.
		3.2.4.1.3-12 The ABDAR system shall provide access to information on fasteners.
		3.2.4.1.3-13 The ABDAR system shall provide access to stress information materials.
		3.2.4.1.3-16 The ABDAR system shall provide drill-size for an identified rivet.
		3.2.4.3-2 The ABDAR system shall present technical data required to accomplish or verify input conditions (as defined in MIL-D-87269) are met.
		3.2.4.3-3 The ABDAR system shall present technical data to accomplish repair tasks.
		3.2.4.3-6 The ABDAR system shall provide technical data and process controls needed to inspect repairs and document inspections.
		3.2.4.3-9 The ABDAR system shall provide data required to accomplish or verify that post conditions are met.

Package	Description	Requirements
		3.2.4.3-12 The ABDAR system shall provide technical data needed to prepare an A/C for disposition.
		3.2.6.2-3 The ABDAR system shall provide access to all weapon system technical data.
		3.2.6.2-4 The ABDAR system shall access the general ABDR TO (1-1H-39)>
		3.2.6.2-12 The ABDAR system shall provide the capability to view technical data by zone or by system.
17.2 Package 2	3.2.2.2 Interview Aircrew Function	3.2.2.2-2 The ABDAR system shall provide debrief information for appropriate databases.
	3.2.4.1.1 Damage Site Location and Triage Sub-Function	3.2.4.1.1-6 The ABDAR system shall provide a way to annotate drawings of the A/C or components that are maintained in the system database.
	3.2.4.1.2 Damage Evaluation Sub-Function	3.2.4.1.2-4 The ABDAR system shall implement a capability to identify and assess individual damages within a damage site.
		3.2.4.1.2-11 The ABDAR system shall use a standard methodology for identifying location of surfaces, components, and parts. Once a location system is decided upon for the ABDAR system, it shall be used consistently.
		3.2.4.1.2-14 The ABDAR system shall provide the MESL.
		3.2.4.1.2-19 The ABDAR system shall contain a capability to identify and evaluate critical A/C subsystems using the MESL and system attributes coupled with damage identification.
		3.2.4.1.2-22 The ABDAR system shall provide access to UXO sample pictures for reference.
		3.2.4.1.2-23 The ABDAR system shall provide examples of types of damage caused by different types of ordnance.

Package	Description	Requirements
		3.2.4.1.2-28 The ABDAR system shall provide the capability to identify and assess damage resulting from hazard migration as a result of ABD.
		3.2.4.1.2-29 The ABDAR system shall provide the capability to predict the impact of damage to the remainder of the system, or integrating systems.
		3.2.4.1.2-37 The ABDAR system shall provide the capability to add notations on the pictures and graphics being used to document the damage.
		3.2.4.1.2-38 The ABDAR system shall provide a capability for the engineer to debrief the assessor/technician.
	3.2.4.1.3 Design Damage Repairs Sub-Function	3.2.4.1.3-1 The ABDAR system shall provide a list of all repair options capable of restoring the A/C that are available for each damage.
		3.2.4.1.3-40 The ABDAR system shall provide the engineer access to information in the ABDR System Program Office (SPO) Engineering Handbook.
	3.2.5.2 Document Repairs Function	3.2.5.2-1 The ABDAR system shall provide for collecting and storing documentation for accomplishment of a repair task. Documentation will include discrepancy and repair actions, operational checks required, and periodic inspections required.
		3.2.5.2-2 The ABDAR system shall document completion of Quality Inspections/In-Process Inspections at appropriate steps in the repair sequence.
		3.2.5.2-6 The ABDAR system shall provide a capability to save text documents (log books).
	3.2.6.2 ABDAR System General Support Function	3.2.6.2-9 The ABDAR system shall provide ability to convert fractions to decimals.

Package	Description	Requirements
		3.2.6.2-11 The ABDAR system shall provide the capability to perform text searches through documentation developed during ABDR of an A/C.
17.3 Package 3	3.2.4.1.1 Damage Site Location and Triage	3.2.4.1.1-7 ABDAR system shall provide guidelines for identification of projectile type (based on entry damage).
	3.2.4.1.2 Damage Evaluation Sub-Function	3.2.4.1.2-9 ABDAR shall provide a trajectory tracing aid.
		3.2.4.1.2-10 ABDAR system shall incorporate a capability to identify and locate collateral damage.
		3.2.4.1.2-16 ABDAR system shall provide the means to recognize and evaluate external indications of internal damage, such as skin buckling and fastener shear.
		3.2.4.1.2-20 ABDAR system shall provide expanded suggestions for locating damage areas if an entry hole exists, but no exit.
		3.2.4.1.2-21 ABDAR system shall provide guidelines for evaluating the risk of any UXO present, given type of entry damage.
		3.2.4.1.2-24 ABDAR system shall provide the capability and associated info necessary to assess fire damage.
		3.2.4.1.2-32 ABDAR system shall provide access to operator's manuals on repair/test equipment.
		3.2.4.1.2-39 ABDAR system shall present data needed to assist the engineer in examining an out-of - limits damage site.
	3.2.4.1.3 Design Damage Repairs	3.2.4.1.3-3 ABDAR system shall provide the capability to generate ETICs and ETTRs.
		3.2.4.1.3-4 ABDAR system shall provide the ability to show the cure time, separate from the ETTR.

Package	Description	Requirements
		3.2.4.1.3-5 ABDAR system shall provide ability to revise ETTRs and ETICs based on work stoppages/slow downs.
		3.2.4.1.3-6 ABDAR system shall provide the ability to generate an Estimated Time To Complete (ETTC) for just the assessment portion of the effort.
		3.2.4.1.3-7 ABDAR system shall provide a method to modify standard repairs (I.e., change tech data).
		3.2.4.1.3-9 ABDAR system shall provide a info on adjacent areas which may be affected by rivet placement.
		3.2.4.1.3-19ABDAR system shall provide the assessor info on special tool requirements.
		3.2.4.1.3-20 ABDAR system shall provide the engineer the capability to design repairs. This capability may include finite-element analysis, automated function calculations, engineering drawings, safety tolerances, material and fastener properties, etc)
		3.2.4.1.3-21 ABDAR system shall provide the capability to calculate stress equations.
		3.2.4.1.3-22 ABDAR system shall provide the capability to calculate bending equations.
		3.2.4.1.3-23 ABDAR system shall provide the capability to calculate compression/buckling equations.
		3.2.4.1.3-24 ABDAR system shall provide the capability to show the assumptions about all equations.
		3.2.4.1.3-25 ABDAR system shall provide capability to calculate center of gravity and lateral symmetry equations.

Package	Description	Requirements
		3.2.4.1.3-26 ABDAR system shall provide the ability to incorporate safety factors into the designed repairs.
		3.2.4.1.3-27 ABDAR system shall provide access to info on primary and secondary load paths.
		3.2.4.1.3-28 ABDAR system shall provide the ability to plot mathematical functions.
		3.2.4.1.3-29 ABDAR system shall provide capability to calculate drag and weight factors.
		3.2.4.1.3-30 ABDAR system shall provide access to local points of contact information.
		3.2.4.1.3-34 ABDAR system shall provide access to a checklist that will walk through repair design procedures.
		3.2.4.1.3-35 ABDAR system shall provide for assisting the engineer in graphically generating an engineering repair design.
		3.2.4.1.3-36 ABDAR system shall identify appropriate repair tasks for identified ABD.
		3.2.4.1.3-37 ABDAR system shall provide for assisting the engineer in generating engineering clarifications.
		3.2.4.1.3-38 ABDAR system shall provide metal substitution information beyond that provided in TO data.
		3.2.4.1.3-39 ABDAR system shall provide the engineer access to info in MIL-HDBK-5.
		3.2.4.1.3-41 ABDAR system shall provide the engineer the capability to reference air frame contractor handbooks.
		3.2.4.1.3-43 ABDAR system shall provide the capability to calculate moment of inertia equations.

Package	Description	Requirements
		3.2.4.1.3-44 ABDAR system shall provide a graphical means of attaching notes to previously designed repairs so they can be re-used on a new damage instance.
		3.2.4.1.3-45 ABDAR system shall provide the engineer the capability to establish editing authority on repair designs.
	3.2.4.2 Repair Planning Function	3.2.4.2-1 ABDAR system shall provide tools for generating repair plans. The tools include a repair plan template that aids the assessor in sequencing repairs, assigning personnel and other resources to each repair, and tracking progress.
		3.2.4.2-2 ABDAR system shall provide the assessor the means to edit estimated repair times based upon actual conditions which currently prevail.
		3.2.4.2-3 ABDAR system shall provide a capability to assist the assessor in sequencing tasks in the repair plan
		3.2.4.2-4 ABDAR system shall provide for modifying a damage site repair plan.
		3.2.4.2-5 ABDAR system shall select and display appropriate methods for inspecting repairs.
		3.2.4.2-6 ABDAR system all provide for creating, recording, and displaying inspection requirements identified by assessors and engineers.
		3.2.4.2-7 ABDAR system shall provide the assessor the capability to incorporate the engineers repair designs into the damage site repair plan.
	3.2.4.3 Perform ABD Repair Function	3.2.4.3-10 ABDAR system shall provide access to an engine run checklist.

Package	Description	Requirements
		3.2.4.3-11 ABDAR system shall provide access to the "Save List" for no-fix A/C.
		3.2.4.3-13 ABDAR system shall provide the capability to sign off on repairs (e.g., Clear a Red X).
	3.2.5.1 Document Assessment Function	3.2.5.1-3 ABDAR system shall provide for signing off 'UXO clear'.
	3.2.6.1 ABDAR System Control Function	3.2.6.1-7 ABDAR system shall provide capability to exchange info at shift change.
	3.2.6.2 ABDAR System General Support Function	3.2.6.2-6 ABDAR system shall provide the capability to annotate checklists.
		3.2.6.2-7 ABDAR system shall provide a set of personal files or personal work region.
		3.2.6.2-8 ABDAR system shall provide the capability to do common text editing and simple drawings.
		3.2.6.2-10 ABDAR shall provide access to "general knowledge" info.
		3.2.6.2-13 ABDAR system shall provide methods of indicating dangerous areas in, on, and around the A/C.
		3.2.6.2-15 ABDAR system shall provide instructions on tool use.
		3.2.6.2-16 ABDAR system shall provide access to info pertaining to MOPP requirements.
		3.2.6.2-17 ABDAR shall provide a map of the base where ABDR duties are being performed.
		3.2.6.2-18 ABDAR system shall provide definitions of Engineering and Maintenance terminology.
		3.2.6.2-19 ABDAR system shall provide access to info that is gathered through job experience such as "Tricks of the Trade".
		3.2.6.2-23 ABDAR system shall provide chemical suit and mask instructions.

Package	Description	Requirements
		3.2.6.2-24 ABDAR system shall provide a means to display threat conditions and MOPP levels.
		3.2.6.2-25 ABDAR system shall provide the capability to reference threat instructions.
		3.2.6.2-29 ABDAR system shall provide Self-Aid/Buddy Care info.
		3.2.6.2-32 ABDAR system shall provide access to info pertaining to local EPA/HAXMAT requirements.
		3.2.6.2-33 ABDAR system shall provide MSDS info.
	3.3.1.3 Aircraft Interface	3.3.1.3-1 ABDAR system shall interface with on-aircraft maintenance buses to provide access to on-board computers, systems, sensors, and maintenance information.
		3.3.1.3-2 ABDAR system shall provide the ability to analyze on-board historical data obtained from the A/C interface.
		3.3.1.3-3 ABDAR system shall provide capability to upload configuration data across the A/C interface.
		3.3.1.3-4 ABDAR system shall provide ability to extract data from DTCs.
17.4 Package 4	Minimum (#1s)	3.2.2.2-1 The ABDAR system shall implement a capability to receive and use info from pilot call-in to begin ABDR process.
		3.2.3.1-4 The ABDAR system shall provide ABDAR Tool and Materials Kit inventory including location of items within the kit.
		3.2.3.1-13 The ABDAR system shall provide for tracking tool check-out/check-in and tool accountability.
		3.2.3.1-23 Track tool availability.
		3.2.4.3-5 ABDAR system shall assist the technician in obtaining assessor clarification of technical data.

Package	Description	Requirements
		3.2.5.2-4 The ABDAR system shall provide the capability to collect and store historical data for long term storage and analysis and ultimate transfer to CAMS/TICARRS/ SURVIAC.
	3.2.2.2 Interview Aircrew Function	3.2.2.2-3 The ABDAR system shall provide means of sending/receiving an assessor assignment to debrief an aircrew, including location and time of the debrief.
		3.2.2.2-4 Using info from the debrief, the ABDAR system shall identify equipment/material needed for initial assessment activities.
	3.2.3.1 Manage Maintenance Resources Function	3.2.3.1-1 The ABDAR system shall implement a capability to levy a resource request on an external organization.
		3.2.3.1-2 The ABDAR system shall provide part-availability info whenever a part-ordering process is initiated.
		3.2.3.1-3 The ABDAR system shall implement a capability to record acceptance and accountability of resources obtained from external sources.
		3.2.3.1-5 The ABDAR system shall provide personnel qualifications in describing personnel resources.
		3.2.3.1-6 The ABDAR system shall provide the capability to establish ordering authority from supply.
		3.2.3.1-7 The ABDAR system shall provide access to ABDR team deployment info w/o displaying the entire Time Phased Force Deployment List.
		3.2.3.1-8 The ABDAR system shall provide tracking local bench stock.
		3.2.3.1-9 The ABDAR system shall provide access to parts availability info.

Package	Description	Requirements
		3.2.3.1-10 The ABDAR system shall provide access to external organization materiel availability info.
		3.2.3.1-14 Track broken tools.
		3.2.3.1-15 Track lost tool incidents.
		3.2.3.1-19 Implement capability to request equipment.
		3.2.3.1-20 Track facility availability.
		3.2.3.1-21 Locate available facilities.
		3.2.3.1-22 Track personnel availability.
		3.2.3.1-23 Track tool availability.
		3.2.3.1-24 Track part status of each cannibalized A/C.
		3.2.3.1-25 Capability to receive a status update of parts on order from supply.
		3.2.3.1-26 Capability to automatically order needed parts from supply.
		3.2.3.3-8 ABDAR system shall provide assessor with capability to assign assessment resources (personnel, equipment, and tools) to damage sites.
		3.2.3.3-9 ABDAR system shall provide for assigning resources (personnel, equipment, parts, and tools) to a repair.
		3.2.3.3-10 ABDAR system shall provide assessor with means to evaluate the availability of repair materials and parts.
		3.2.3.3-11 ABDAR system shall provide the assessor the means to evaluate the availability of facilities.
		3.2.3.3-12 ABDAR system shall provide the assessor the mains to evaluate the availability of personnel.
		3.2.3.3-13 ABDAR system shall provide the assessor the means to evaluate availability of support equipment.

Package	Description	Requirements
	3.2.4.1.1 Damage Site Location and Triage	3.2.4.1.1-3 ABDAR system shall provide the capability to send a warning notification of any UXO found on the A/C to the ABDAR system server and to receive and display such a warning from the server.
		3.2.4.1.1-4 ABDAR system shall provide capability to initiate request for EOD personnel to remove UXO.
	3.2.4.1.2 Damage Evaluation Sub-Function	3.2.4.1.2-1 ABDAR system shall provide aids for managing assessment resources.
		3.2.4.1.2-8 ABDAR system shall provide the assessor the capability to request engineering assistance for clarification of degree and extent of damage.
		3.2.4.1.2-25 ABDAR system shall provide capability to interface with the MIL-STD 1553 data bus in order to download system information.
	3.2.4.1.3 Design Damage Repairs	3.2.4.1.3-14 ABDAR system shall provide access to historical info about previous repairs on the A/C.
		3.2.4.1.3-18 ABDAR system shall provide capability to input a part number so the part can then be viewed on a diagram.
		3.2.4.1.3-17 ABDAR system shall provide the engineer access to weapon system engineering drawings for the entire A/C.
		3.2.4.1.3-31 ABDAR system shall provide the capability to communicate with Airframe Contractors.
		3.2.4.1.3-32 ABDAR system shall provide the ability to communicate with Depot.
		3.2.4.1.3-33 ABDAR system shall provide the ability to communicate with off-site engineers.

Package	Description	Requirements
		3.2.4.1.3-42 ABDAR system shall provide the engineer capability to reference selected textbooks.
	3.2.6.1 ABDAR System Control Function	3.2.6.1-1 ABDAR shall provide the capability to manage and support multiple, simultaneous assessments and repairs.
		3.2.6.1-3 ABDAR system shall maintain the current state of all A/C in ABDAR maintenance.
		3.2.6.1-4 ABDAR system shall provide tools to ensure all assessment and repair activities are complete prior to preparing the A/C for disposition.
		3.2.6.1-5 ABDAR system shall provide a mechanism to stop all assessment and repair activities when the assessor determines further actions are not warranted.
		3.2.6.1-8 ABDAR system shall provide the ability to communicate with other members of the ABDR team.
	3.2.6.2 ABDAR System General Support Function	3.2.6.2-1 ABDAR system shall provide a capability that will allow authorized users to update/edit/modify any of the locally variable files (certification rosters, inventories, personnel rosters, etc.).
		3.2.6.2-21 ABDAR system shall provide a means to create load lists, packing lists, and required forms.
		3.2.6.2-26 ABDAR system shall provide the capability to notify the appropriate authorities of an emergency condition.
		3.2.6.2-27 ABDAR system shall provide documentation of injury information.
		3.2.6.2-28 ABDAR system shall provide access to info pertaining to Air Force Occupational Safety and Health (AFOSH) standards.

Package	Description	Requirements
		3.2.6.2-30 ABDAR system shall provide the capability to produce an audit trail.
		3.2.6.2-31 ABDAR system shall provide access to Maintenance operating Instructions (MOIs)/Standard Operating Procedures (SOPs).
17.5 Package 5	Integrated TOs	
	Upgrade ABDAR software to use an integrated digital TO.	(See TO Req. for Package 1)
	Develop software to parse a PDF version of a weapon system ABDR TO and store into the ABDAR database for use by the application.	